Fair Routing for Resilient Packet Rings

LI Cheng

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

In

Information Engineering

November 2002

Abstract

In this thesis, we investigate the routing and flow control problems for Resilient Packet Ring (RPR) network. By default, shortest path based routing scheme is used for RPR network. However, since the traffic load distributes unevenly, this nonflexible routing scheme is not always efficient. This motivates us to look for new routing and flow control schemes which can utilize bandwidths efficiently. Given the link capacities and traffic demands, we design an optimal routing and flow control algorithm to maximize throughput. Numerical results show that the throughput is improved by optimal routing compared to shortest path routing. We also propose the *Progressive Filling with Optimal Routing* (*PFOR*) algorithm to solve max-min fair allocation and optimal routing problems. Numerical results show that the throughput is improved by *PFOR* compared to traditional *Progressive Filling* (*PF*) algorithm. Finally, we study the tradeoff between throughput and fairness. By properly controlling the amount of the tradeoff, throughput can gain much at only a little sacrifice of fairness.

Although our work focuses on RPR network, the results can be generalized to any network with a mesh topology.

摘 要

在這篇論文中, 我們研究 Resilient Packet Ring (RPR) 網絡的路由和流量控制問題. RPR 網絡默認的路由方式是最短路徑優先方式. 但是, 由於流量負載的非均勻分布, 這種不靈活的路由方式不能有效的利用帶寬. 這驅使我們尋找一個能有效利用帶寬的路由和流量控制方式. 在鏈路容量和流量需求已知的條件下, 我們從吞吐量最大化角度設計了最優選路算法. 結果顯示, 同最短路徑優先方式相比, 最優選路方式提高了網絡的吞吐量. 我們還提出了 *Progressive Filling with Optimal Routing* (*PFOR*) 算法來同時解決帶寬公平分配和最優選路問題. 結果顯示, 同傳統的 *Progressive Filling* (*PF*) 算法相比, *PFOR* 提高了網絡的吞吐量. 最後, 我們研究了吞吐量與帶寬公平分配的折衷問題. 通過適當的調整折衷的程度, 我們可以以犧牲較少的公平性的代價來換取更大的吞吐量.

雖然我們得到的結果是基于 RPR 網絡模型, 但我們的結果也適用於一般的 mesh 結構的網絡模型。

Acknowledgement

I would like to express my warmest gratitude toward my research supervisor, Prof. Tak-shing Peter, Yum, for his support in my research. Without his patient guidance, I would not taste the fun in *Networking*.

Specially, I would thank Prof. Eric Wong, for teaching me how to enjoy my job.

Thanks to Yang Yang, Hua Cunqing, Nie Xiaofeng, for their kindly help.

Contents

CHAPTER 1 INTRODUCTION 1

1.1 The Evolution of Ring Network Technologies 1

1.1.1 Token Ring Technology 1

1.1.2 Resilient Packet Ring Technology 4

1.2 Optimal Routing 7

1.3 Fairness 8

1.4 Outline of Thesis 10

CHAPTER 2 OPTIMAL ROUTING 11

2.1 Throughput Analysis 11

2.2 Numerical Results 13

CHAPTER 3 OPTIMAL FAIR ROUTING 19

3.1 Overview 19

3.2 Max-min Fair Allocation 19

3.3 Proportionally Fair Allocation 32

3.4 Numerical Results 33

CHAPTER 4 TRADEOFF ANALYSIS 40

4.1 Tradeoff between Throughput and Max-min Fairness 40

4.2 Numerical Results 42

4.3 Tradeoff between Throughput and Utility 47

4.4 Numerical Results 48

CHAPTER 5 CONCLUSION 54

5.1 Summary 54

5.2 Discussion and Future Work 55

BIBLIOGRAPHY 57

List of Figures

FIGURE 1.1 RESILIENT PACKET RING NETWORK TOPOLOGY 6

FIGURE 1.2 SPATIAL REUSE CAPABILITY OF RESILIENT PACKET RING NETWORK 6

FIGURE 2.1 THE 4-NODE PACKET RING NETWORK 17

FIGURE 2.2 THROUGHPUT IMPROVEMENT OF THE MAXIMUM THROUGHPUT ROUTING 17

FIGURE 2.3 MAXIMUM CARRIED LOAD IMPROVEMENT BY OPTIMAL ROUTING OVER SHORTEST PATH ROUTING 18

FIGURE 3.1 THE FLOWCHART OF *PFOR* 28

FIGURE 3.2 4-NODE PACKET RING NETWORK EXAMPLE 29

FIGURE 3.3 STEP 1 OF *PFOR* EXAMPLE 29

FIGURE 3.4 STEP 2 OF *PFOR* EXAMPLE 30

FIGURE 3.5 STEP 3 OF *PFOR* EXAMPLE 30

FIGURE 3.6 ANOTHER FLOW PARTITIONING SOLUTION OF *PFOR* EXAMPLE 31

FIGURE 3.7 AN EXAMPLE OF *PF* WITH SHORTEST PATH ROUTING 31

FIGURE 3.8 4-NODE PACKET RING NETWORK 37

FIGURE 3.9 THROUGHPUT VS. OFFERED LOAD FOR MAX-MIN FAIR ALLOCATION 37

FIGURE 3.10 THROUGHPUT VS. OFFERED LOAD FOR PROPORTIONALLY FAIR ALLOCATION 38

FIGURE 3.11 UTILITY VS. OFFERED LOAD FOR PROPORTIONALLY FAIR ALLOCATION 38
FIGURE 3.12 THROUGHPUT IMPROVEMENT FOR MAX-MIN FAIR ALLOCATION AND PROPORTIONALLY FAIR ALLOCATION OVER OPTIMAL ROUTING 39
FIGURE 4.1 THROUGHPUT VS. OFFERED LOAD 44
FIGURE 4.2 FAIRNESS INDEX VS. OFFERED LOAD 44
FIGURE 4.3 THROUGHPUT VS. THE PARAMETER *B* 45
FIGURE 4.4 FAIRNESS INDEX VS. THE PARAMETER *B* 45
FIGURE 4.5 THROUGHPUT GAIN VS. FAIRNESS SACRIFICE 46
FIGURE 4.6 THROUGHPUT VS. OFFERED LOAD 51
FIGURE 4.7 THE AGGREGATE UTILITY VS. OFFERED LOAD 51
FIGURE 4.8 THROUGHPUT VS. *B* 52
FIGURE 4.9 THE AGGREGATE UTILITY VS. *B* 52
FIGURE 4.10 THROUGHPUT GAIN VS. THE AGGREGATE UTILITY SACRIFICE 53

List of Notations

N	The number of nodes
C_i	The capacity of the ith link in the outer ring
D_i	The capacity of the ith link in the inner ring
u_i	The load of the ith link in the outer ring
v_i	The load of the ith link in the inner ring
b	The tuning parameter
$\boldsymbol{\gamma} = \left[\gamma_{ij}\right]_{N\times N}$	Traffic matrix (Demand)
$\mathbf{x} = \left[x_{ij}\right]_{N\times N}$	The flow rate matrix
$\mathbf{y} = \left[y_{ij}\right]_{N\times N}$	The outer ring rate matrix
$\mathbf{z} = \left[z_{ij}\right]_{N\times N}$	The inner ring rate matrix
$\mathbf{P}_{\mathrm{b}} = \left[P_{bij}\right]_{N\times N}$	The blocking probability matrix
$\boldsymbol{\alpha} = \left[\alpha_{ij}\right]_{N\times N}$	The percent (percent of traffic crossing the outer ring) matrix
$\mathbf{E} = \left[e_{ij}\right]_{N\times N}$	The marked flow rate matrix
$\mathbf{H} = \left[h_{ij}\right]_{N\times N}$	The flow rate matrix of max-min fair allocation

Chapter 1

Introduction

Ring networks were considered for use for Local Area Networks (LANs), Metro Area Networks (MANs) and Wide Area Networks (WANs) for a long time, due to their low complexity and simple topology structure. Usually, Token Ring technology is used for LANs. This technology is not efficient for use in a MAN or WAN, because the ratio between the propagation delay and the transmission delay is large. MANs typically utilize SONET/SDH ring technology. However, with the volume of data traffic in metro networks growing, SONET/SDH ring networks cannot transmit data traffic efficiently, because they are optimized for voice and circuit-switched traffic. To meet the demands for metropolitan IP-optimized networks, Resilient Packet Ring (RPR) technology has been proposed [2]. In this chapter, we first give a brief review of the evolution of ring networks. Then we introduce the RPR technology.

1.1 The Evolution of Ring Network Technologies

1.1.1 Token Ring Technology

The Token Ring technology was originally developed by IBM. Later, it was included as the IEEE 802.5 Token Ring LAN standard [39] by IEEE. In a Token

Ring network, a small frame, called a token, circulates around the whole ring whenever all nodes are idle for admission control. If a node receiving the token has no data to transmit, it just passes the token to the next node. When a node wants to transmit data, it is required to get the token and remove it from the ring before transmission. Each node may hold the token for a maximum period of time. When the information frame is circulating in the ring, no token is on the ring. So other nodes which have data to transmit must wait. When the information frame gets to the destination node, the destination node copies the information frame. Then the information frame continues to move along the ring and is finally removed by the sending node. Then a new token is generated by the node and circulates around the ring.

The IEEE 802.5 Token Ring LANs utilize copper wire. For low speeds and short distances, this will work well. However, for high speeds and long distances, LANs must utilize the fiber optics technology. For this purpose, FDDI (Fiber Distributed Data Interface) was proposed and later included as the ANSI X3 standard [40]. FDDI is a fiber optic token ring LAN using multimode fibers. A FDDI backbone consists of two counter-rotating fiber rings, one transmits data clockwise, while the other counterclockwise. At any moment, only one ring works and the other is used as a backup. The basic FDDI protocols are similar to IEEE 802.5, the Token Ring protocols. To transmit data, a node must first get the token. Then it transmits an information frame and removes it when it returns and then generates a new token.

IEEE 802.5 Token Ring LAN and FDDI are congestion-free because there is only one token and only one node can transmit data at a moment. However, this congestion-free feature is achieved at the expense of poor bandwidth utilization.

Especially, for a high speed and large span ring, packet transmission time is much shorter than the propagation delay. Under such conditions, the performance of the traditional token ring network degrades. To improve the bandwidth utilization, many spatial reuse (the ability to provide concurrent transmission over distinct segments of the ring) technologies have been proposed. Generally, they can be classified into two groups. One is Early Token Release (ETR) media access control [41] and the other is multi-token media access control [42].

Concurrent Transmission Ring (CTRing) [43], T-S Ring [44] and Pipeline Ring [45] are based on ETR technology. Instead of source stripping (the token is removed by the transmitting node), ETR uses destination stripping (the token is removed by the receiving node). Once an information frame gets to its destination, it is removed by the destination node. The destination node then generates a special "conditional token". It indicates that a portion of the ring, but not the whole ring, is available. Each node which has data to transfer can check the source-destination address information in the "conditional token" to know which part of the ring is currently in use. A node is free to transmit data if the portion of the ring between itself and its destination is unused by other nodes. In this way, there may be multiple packets propagating on the ring at the same time. Therefore, the bandwidth utilization is improved.

Another technique for improving bandwidth utilization is to use multiple tokens so that the chance for a node to get a token increases. One protocol proposed by [46] and the protocol used in MetaRing [42] network are based on the multi-token technology. It is assumed that there are at most M (M>1) tokens on the ring at the same time. Tokens may be generated by one or more nodes. For the case that there is only one startup node, multiple tokens may be generated by equally spaced time interval. The other case is that there are multiple startup nodes which can generate tokens independently. Basically, the token seizing and releasing scheme is similar to that of the single Token Ring technology. One difference is that if a token arrives at a transmitting node, the arriving token will be discarded. By using multiple tokens, there may be more than one transmitting nodes and the bandwidth is spatially reused.

1.1.2 Resilient Packet Ring Technology

RPR consists of two counter-rotating rings. Each ring segment can be used independently to pass both data and control packets. Control packets propagate in the opposite direction from the corresponding data packets. One of the rings is referred to as the inner ring and the other as the outer ring as illustrated in Figure 1.1. Unlike SONET/SDH ring networks which reserve half of the rings for protection, both the inner and the outer ring are used concurrently, allowing the service provider to increase bandwidth usage.

FDDI and Token Ring [4] use source stripping and tokens to control ring access. Packets circulate around the entire ring before being stripped by the source. In contrast, RPR performs destination stripping of packets. Since nodes may concurrently transmit packets without waiting for a shared token, different ring segments can be simultaneously used. An example is illustrated in Figure 1.2. In this case, there are three active flows passing through the outer ring. They are flow (4, 1), flow (5, 0) and flow (2, 3), respectively. Flow (4, 1) and flow (5, 0) share the bandwidth of link (5, 0). Flow (2, 3) owns the whole bandwidth of link (2, 3) at this moment. On the contrary, for FDDI or Token Ring, the whole ring bandwidth is owned by only one flow at any moment, thus more resource is wasted.

Another feature of the RPR technology is robust resiliency and restoration. It uses intelligent protection switching (IPS) to provide proactive performance monitoring, rapid self-healing and IP service restoration after ring or fiber faults. Basically, IPS is similar to automatic protection switching (APS) [5] used in SONET/SDH networks. Unlike APS, IPS provides an additional set of packet-optimized capabilities. IPS is discussed in detail in [1].

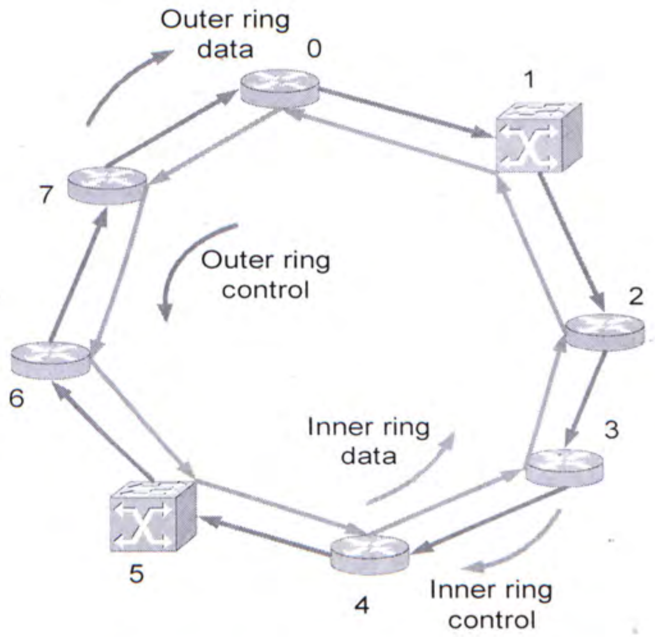

Figure 1.1 Resilient Packet Ring network topology

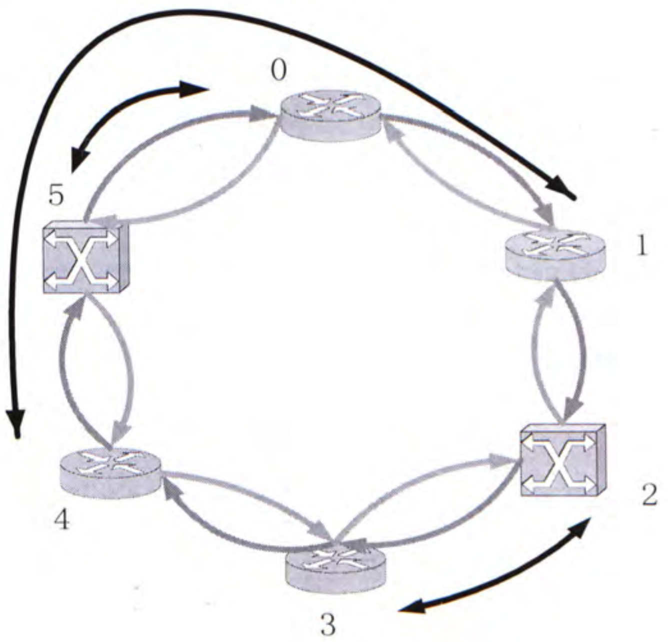

Figure 1.2 Spatial Reuse capability of Resilient Packet Ring network

1.2 Optimal Routing

Traditionally, many routing algorithms are designed based on the shortest path routing, such as Bellman-Ford algorithm [6] and Dijkstra algorithm [7]. A shortest path routing algorithm transfers each packet along a minimum length path between the origin and the destination nodes. The simplest scenario treats each link to have unit length. In this case, a shortest path is simply a path with minimum number of links (also called a minimum-hop path). The shortest path routing scheme is widely used in MAN and WAN as well as RPR.

IP traffic may change dynamically and distribute unevenly within a subnet. By using a single route, the shortest path routing cannot utilize the bandwidth efficiently because of its inflexibility. The shortest path routing has two drawbacks [7]. First, it uses only one path per source-destination pair, so it potentially limits the throughput of the network. Second, its capability to adapt to changing traffic conditions is limited. Optimal routing, based on the optimization of a particular objective, such as delay or throughput, can eliminate both of these disadvantages by splitting source-destination pair traffic at the source and by changing traffic gradually between the two feasible paths.

We consider the following example that maximizes the throughput for the packet ring network shown in Figure 1.2. We assume that all links have the same capacity of 100 units per second and the demands of flow (4, 1), (5, 0) and (2, 3) are 80, 70 and 50 units per second, respectively. If the shortest path routing is used, some packets either in flow (4, 1) or flow (5, 0) will be dropped, since the link between node 5 and node 0 is a bottleneck. The blocking probability and the total throughput of the network are 25% and 150 units per second, respectively.

However, if we route 50 percent of traffic of flow (4, 1) to an alternative path or go through the inner ring, then no traffic will be blocked and the total throughput is 200 units per second.

This example shows that optimal routing can improve the bandwidth efficiency compared to the shortest path routing. In a later chapter, we will formulate an optimal routing problem that maximizes throughput for a packet ring network and show the improvement of the throughput performance by numerical results.

1.3 Fairness

Since a packet ring network is shared by many parties as a backbone, fairness in bandwidth allocation may be necessary, especially when the network is under high offered load conditions. Many fairness criteria have been studied in the literature [16], [17]. Nowadays, two fairness criteria are widely adopted. One is *max-min fairness* [18] and the other is *proportional fairness* [12], [15].

The concept of *max-min fairness* is described as follows. We assume that a subnet is shared by a set of flows $\boldsymbol{P}$. We denote by r_p the allocated rate for flow p. Then *max-min fairness* can be defined as follows [7].

***Definition 1*:** A vector of rates $\boldsymbol{r}$ is said to be *max-min fair* if it is feasible and for each $p \in \mathbf{P}$, r_p can not be increased while maintaining feasibility without decreasing $r_{p'}$ for some flow $p' \in \mathbf{P}$ for which $r_{p'} \le r_p$, $r_p, r_{p'} \in \mathbf{r}$

The name "*max-min*" comes from the idea that it is forbidden to decrease the share of sources that have small values. Thus, we give priority to flows with small rates. The theorem [7] below tells us how to obtain a max-min fair allocation within a particular network.

Theorem 1: A feasible rate vector $\boldsymbol{r}$ is *max-min fair* if and only if each flow has a bottleneck link with respect to $\boldsymbol{r}$.

The proof can be found in [20]. This theorem is useful for us to derive practical methods to achieve max-min fair allocation. Based on this theorem, we derive a frame work to achieve max-min fair allocation for general network allowing optimal routing. The detail of this algorithm will be discussed in a later chapter.

The concept of *proportional fairness* as follows [12].

Definition 2: A feasible rate vector $\boldsymbol{x}$ is *proportionally fair* if and only if, for any other feasible rate vector $\boldsymbol{y}$, we have:

$$\sum_{s=1}^{S} \frac{y_s - x_s}{x_s} \leq 0 \qquad (1.1)$$

S is the number of users and $x_s \in \mathbf{x}, y_s \in \mathbf{y}$.

In other words, any change in the allocation must have a negative average change. The following theorem [20] tells us how we can achieve a proportionally fair allocation of bandwidth for a particular network.

Theorem 2: There exists one unique proportionally fair allocation. It is obtained by maximizing $J(x) = \sum_s \log(x_s)$ over the set of feasible allocations.

In fact, *proportional fairness* is a special case of a more general utility concept where proportionally fair allocation is the solution of optimizing the aggregate utility.

To evaluate a system's fairness performance, we also need an index or a function that quantifies the fairness. Many indices have been proposed [21]. The most popular one proposed by R. Jain and D. Chiu [22] was defined as follows.

Definition 3: The fairness index is defined as follows:

$$Fairness\ Index = \frac{\left(\sum_i x_i\right)^2}{n\sum_i x_i^2} \tag{1.2}$$

Where n is the number of users and x_i is the normalized throughput of flow i.

Compared with other definitions, this index has the following properties [23]:

- The fairness index is bounded between 0 and 1. An absolute fair allocation has a fairness index 1.
- The fairness index is independent of scale, i.e., unit of measurement does not matter.
- The fairness index is a continuous function.
- If only k of n flows share the resource equally with the remaining n-k flows not receiving any resource, then the fairness index is k/n.

In the following, this fairness index is used to evaluate an allocation's fairness performance.

1.4 Outline of Thesis

In this chapter, we have briefly described the Resilient Packet Ring (RPR) technology. Two important issues, optimal routing and fairness were discussed for this technology.

In Chapter 2, we study the maximum throughput optimal routing problem for a packet ring network. In Chapter 3, we study the problem of optimal routing with fair rate allocation as the objective for a packet ring network. In Chapter 4, we study the tradeoff between throughput and max-min fairness and the tradeoff between throughput and utility. Finally, in Chapter 5, we summarize the thesis and point out possible future work.

Chapter 2

Optimal Routing

2.1 Throughput Analysis

From the point of view of a network operator, we would like to accommodate more traffic in a subnet in order to maximize throughput and hence revenue. This objective can be achieved by properly designing the routing and flow control algorithms. In this chapter, we design an optimal routing and flow control algorithm to maximize the total throughput for a packet ring network. In particular, for the traffic of each source-destination pair, we have to design the distribution of traffic flow between the inner and the outer rings. We assume all link capacities and the demand of each flow are known and no flow is allowed to get more resource than its demand. Then the optimization problem is formulated as follows.

Task. Maximum throughput

Constants:

N	The number of nodes
$\boldsymbol{\gamma} = \left[\gamma_{ij}\right]_{N\times N}$	Given demands for all source-destination pairs
C_i	The capacity of the ith link in the outer ring, $i = 0,1,\cdots,N-1$
D_i	The capacity of the ith link in the inner ring, $i = 0,1,\cdots,N-1$

Variables:

$\mathbf{x} = \left[x_{ij} \right]_{N \times N}$, $i, j = 0,1,\cdots,N-1$ Flow rates

$\mathbf{y} = \left[y_{ij} \right]_{N \times N}$, $i, j = 0,1,\cdots,N-1$ Traffic goes through the inner ring

$\mathbf{z} = \left[z_{ij} \right]_{N \times N}$, $i, j = 0,1,\cdots,N-1$ Traffic goes through the outer ring

Maximize:

$$L(\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{ij} \tag{2.1}$$

Subject to constraints:

$$\begin{aligned} &\sum_{j=(i+1)_{\bmod N}}^{(i-1)_{\bmod N}} z_{ij} + \sum_{k=(i+2)_{\bmod N}}^{(i-1)_{\bmod N}} \sum_{j=(i+1)_{\bmod N}}^{(k-1)_{\bmod N}} z_{kj} \le C_i, \quad i = 0,1,\cdots N-1 \\ &\sum_{j=(i+1)_{\bmod N}}^{(i-1)_{\bmod N}} y_{ij} + \sum_{k=(i+1)_{\bmod N}}^{(i-2)_{\bmod N}} \sum_{j=(k+1)_{\bmod N}}^{(i-1)_{\bmod N}} y_{kj} \le D_i, \quad i = 0,1,\cdots N-1 \end{aligned} \tag{2.2}$$

$$\begin{aligned} z_{ij} - x_{ij} \le 0, &\quad i, j = 0,1,\cdots,N-1 \\ y_{ij} - x_{ij} \le 0, &\quad i, j = 0,1,\cdots,N-1 \end{aligned} \tag{2.3}$$

and

$$\begin{aligned} y_{ij}, z_{ij} \ge 0 &\quad \text{for } i, j = 0,1,\cdots N-1 \text{ and } i \ne j \\ y_{ij}, z_{ij} = 0 &\quad \text{for } i, j = 0,1,\cdots N-1 \text{ and } i = j \end{aligned} \tag{2.4}$$

$$\begin{aligned} 0 \le x_{ij} \le \gamma_{ij}, &\quad \text{for } i, j = 0,1,\cdots,N-1 \text{ and } i \ne j \\ x_{ij} = 0, &\quad \text{for } i, j = 0,1,\cdots,N-1 \text{ and } i = j \end{aligned} \tag{2.5}$$

Constraint (2.2) assures that the aggregate load on a link will not exceed the capacity of the link. It consists of two parts, one for the outer ring and the other for the inner ring. Constraint (2.5) assures that a flow will not get more bandwidths than its demand. The problems of optimal routing and flow control for maximizing throughput can be solved jointly by linear programming techniques.

Therefore, routing solution and bandwidth allocation solution can be obtained simultaneously and directly.

Many algorithms have been developed to solve this Maximum Flow problem with different complexities [47]. We model the network as a directed graph $\mathbf{G}=(\mathbf{N},\mathbf{E})$, where $\boldsymbol{N}$ is the node set and $\boldsymbol{E}$ is the edge set. For simplicity, we say that the network consists of N nodes and E edges. Then Ford-Fulkeson algorithm, Push-Relabel algorithm and Relabel-to-Front algorithm give the complexities $O(NE^2)$, $O(N^2E)$ and $O(N^3)$, respectively [47], [48]. For a packet ring network, we have $E=2N$. Therefore, all three algorithms above give the same complexity $O(N^3)$. The maximum throughput routing problem can be solved in polynomial time.

2.2 Numerical Results

We consider the network shown in Figure 2.1. This packet ring network consists of 4 nodes and 8 links. We assume that all link capacities are the same and equal to 200 units per second. We generate 10 different traffic matrices. Each traffic matrix is used as a base for one experiment. The elements of each matrix are randomly generated in the range [0, 100]. For the case that the offered load index is equal to an integer k (k>0), the corresponding traffic matrix is generated by multiplying each element of the base matrix by k. The results shown in the following are the average values based on ten experiments with different traffic matrices.

Figure 2.2 compares the throughput performance between the maximum throughput routing and the shortest path routing. When the offered load index is equal to 2, the maximum throughput routing increases the throughput by 10.2% compared to the shortest path routing. When the network is under extremely high or low offered load, the maximum throughput routing cannot increase the throughput much. For example, when the offered load index is 10, the throughput for the shortest path routing and the maximum throughput routing are 1486 and 1487 units per second, respectively.

Next, we investigate the results more specifically. We use one of the ten traffic matrices as an example. The traffic matrix γ_4 is (2.6). When the offered load index is equal to 2, the rate allocation solution $\mathbf{X}_4^{'}$ and the flow partition solution $\boldsymbol{\alpha}_4^{'}$ for the shortest path routing are shown in (2.7) and (2.8), respectively. The blocking probability is defined as the ratio of the blocked traffic (the offered load minus the throughput) over the offered load. The overall blocking probability for the shortest path routing is 0.2332 and the blocking probability matrix $\mathbf{P}_{b4}^{'}$ is shown in (2.9), where the element of this matrix represents the blocking probability for a flow. For the shortest path routing, the flow partition solution is always the same as (2.8). The rate allocation solution $\mathbf{X}_4$ and the flow partition solution $\boldsymbol{\alpha}_4$ for the maximum throughput routing are shown in (2.10) and (2.11), respectively. The element of the flow partition matrix represents the percent of the traffic which goes through the outer ring for a flow. The overall blocking probability is 0.1022 and the blocking probability for each individual flow $\mathbf{P}_{b4}$ is shown in (2.12). The maximum throughput routing reduces the overall blocking probability much compared with the shortest path routing. However, for either the shortest path routing or the maximum throughput routing, flow blocking is not uniform, which means that flows are not treated fairly. This may cause severe starvation problem. For example, when the offered load index is equal to 2, flow (0, 2) which has a small demand of 76 (2*38) units per second has a high blocking probability of 0.49 shown in (2.12). In the next chapter, we study the fair bandwidth allocation problem for packet ring networks.

We are also interested in finding the maximum throughput improvement by optimal routing as a function of the number of nodes. We find that the maximum throughput improvement by optimal routing for this 4-node network is about 10%. We further investigate this property for other packet ring networks with 6, 8, 10 and 12 nodes. The corresponding link capacities are 600, 1000, 1400 and 2000 units per second, respectively. The result in Figure 2.3 shows that the larger the network size, the less is the maximum throughput improvement. Therefore, the maximum throughput routing does not improve throughput much for large packet ring networks.

To summarize, the maximum throughput routing improves throughput compared to the shortest path routing. The maximum throughput routing increases the throughput less when a packet ring network becomes larger. The maximum throughput routing cannot treat each individual flow fairly.

$$\gamma_4 = \begin{bmatrix} 0 & 30 & 38 & 50 \\ 19 & 0 & 86 & 90 \\ 19 & 15 & 0 & 82 \\ 68 & 70 & 59 & 0 \end{bmatrix} \tag{2.6}$$

$$\mathbf{X}_4' = \begin{bmatrix} 0 & 60 & 31.08 & 100 \\ 38 & 0 & 149.94 & 18.97 \\ 10.40 & 30 & 0 & 164 \\ 110.81 & 78.79 & 118 & 0 \end{bmatrix} \tag{2.7}$$

$$\boldsymbol{\alpha}_4' = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \tag{2.8}$$

$$\mathbf{P}_{b4}' = \begin{bmatrix} 0 & 0 & 0.59 & 0 \\ 0 & 0 & 0.13 & 0.89 \\ 0.73 & 0 & 0 & 0 \\ 0.19 & 0.44 & 0 & 0 \end{bmatrix} \tag{2.9}$$

$$\mathbf{X}_4 = \begin{bmatrix} 0 & 60 & 38.54 & 92.58 \\ 38 & 0 & 139.31 & 129.57 \\ 38 & 30 & 0 & 164 \\ 136 & 140 & 118 & 0 \end{bmatrix} \tag{2.10}$$

$$\boldsymbol{\alpha}_4 = \begin{bmatrix} 0 & 1 & 0.97 & 0.04 \\ 0 & 0 & 0.99 & 0.16 \\ 0.02 & 0.01 & 0 & 1 \\ 0.94 & 0.50 & 0 & 0 \end{bmatrix} \tag{2.11}$$

$$\mathbf{P}_{b4} = \begin{bmatrix} 0 & 0 & 0.49 & 0.07 \\ 0 & 0 & 0.19 & 0.28 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.12}$$

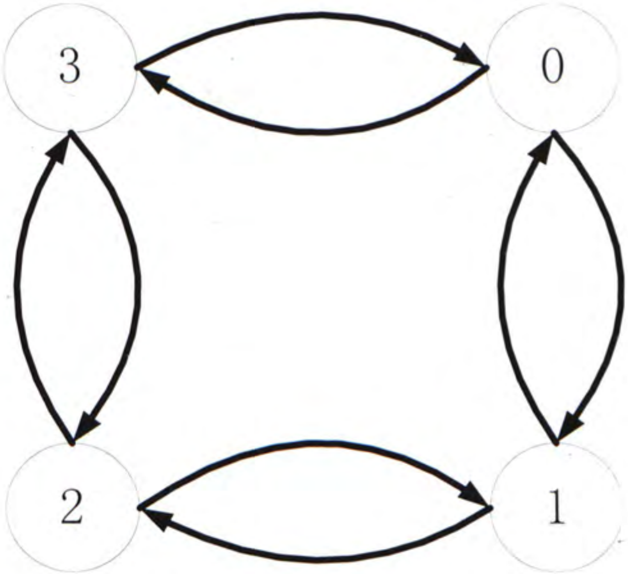

Figure 2.1 The 4-node packet ring network

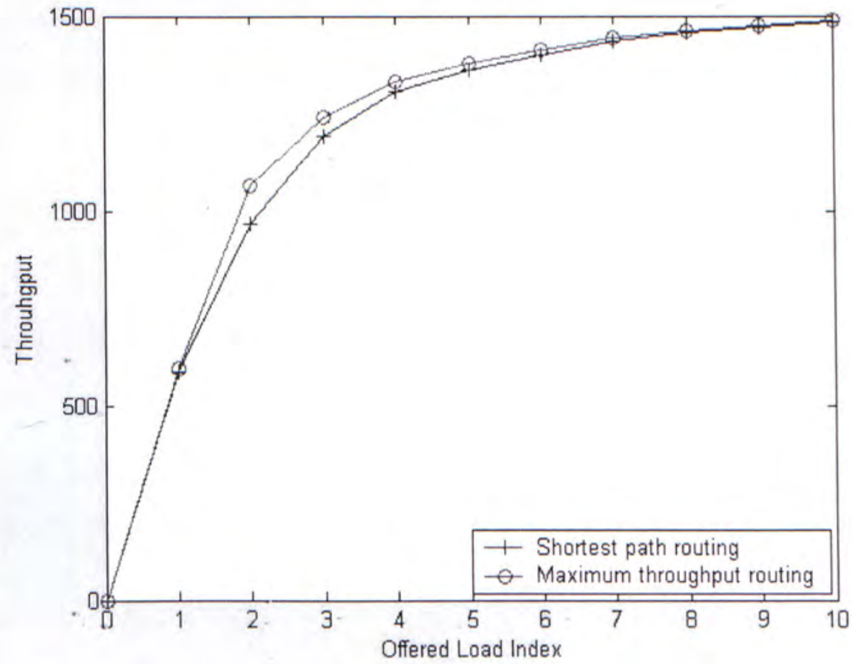

Figure 2.2 Throughput improvement of the maximum throughput routing

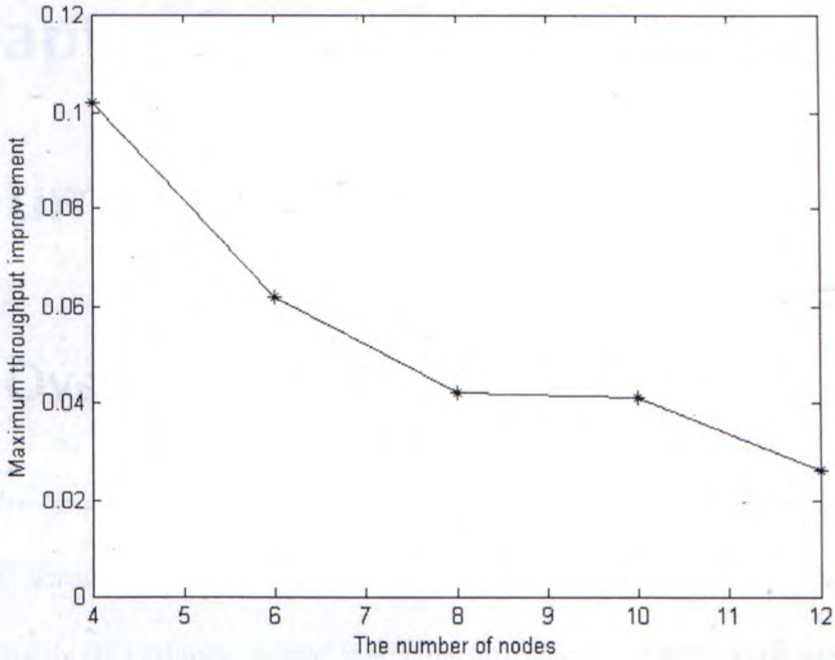

Figure 2.3 Maximum carried load improvement by optimal routing over shortest path routing

Chapter 3

Optimal Fair Routing

3.1 Overview

In the previous chapter, the maximum throughput routing is designed to maximize the total throughput. However, the blocking probability of each individual flow may have high variance. Since many parties share a packet ring network as a backbone, fair bandwidth allocation may be essential. From the point of view of a network operator, fairness should be maintained while sacrificing a little throughput. To achieve this goal, we solve jointly the optimal routing and fair bandwidth allocation problems. In this chapter, we first study the problem of optimal routing with max-min fairness. Then we study the problem of optimal routing with proportional fairness. Finally, we investigate the throughput performance of each fair bandwidth allocation scheme.

3.2 Max-min Fair Allocation

Theorem 1 (Chapter 1) studies that a feasible rate vector $\boldsymbol{r}$ is max-min fair if and only if each flow (a source-destination node pair) has a bottleneck link with respect to $\boldsymbol{r}$ [7]. This theorem is particularly useful in deriving practical methods for obtaining a max-min fair allocation. The algorithm of *Progressive Filling* (*PF*) [20] is derived from this theorem. The general idea of *PF* algorithm is as follows.

We start with all flow rates equal to zero and increase all rates together at the same pace, until one or several link capacity limits are hit. The rates of the flows that use these links are not increased any more. We continue to increase rates of other flows. Each flow whose rate stops growing has at least a bottleneck link. This is because they use a saturated link and all other flows using the saturated link stop increasing their rates at the same time. This algorithm continues until it is impossible to increase the rate for any flow. The algorithm terminates in a finite time because the number of links and flows are finite. The proof that max-min fair allocation can be obtained by the algorithm of *PF* can be found in [20].

However, max-min fair allocation is usually used under the single route assumption. If we use a single path routing scheme, such as the shortest path routing, *PF* algorithm works well to achieve max-min fairness in a general network. However, the result in previous chapter tells us that a single path routing scheme cannot utilize bandwidth efficiently. We argue that max-min fair allocation with optimal routing should give a better throughput performance than max-min fair allocation with shortest path routing. However, to solve max-min fair allocation and routing problems jointly gives us a new challenge. Here, we solve the problems above by proposing a new algorithm, called *Progressive Filling with Optimal Routing* (*PFOR*), which is an extended version of the original *Progressive Filling* (*PF*) algorithm.

PFOR differs from *PF* in the following ways. First, *PF* assumes that all flows have infinite bandwidth demands. On the other hand, *PFOR* considers a more general case that each flow has a finite bandwidth demand and it is not allowed to get more bandwidth than its demand. Second, in *PFOR*, each flow may have several feasible paths. A flow rate stops growing if the flow meets bottleneck links through all its feasible paths or its demand is met. Third and the most important one, in *PFOR*, rate increasing and optimal routing are implemented jointly. In other words, at the end of rate increasing process, each flow rate and its distribution within all its feasible paths are obtained at the same time.

In *PFOR*, we say that a flow is *marked* if its rate is not possible to increase any more. A flow is *marked* in one of the following two cases:

1. A flow gets the bandwidth equal to its demand.
2. A flow meets bottleneck links through all its feasible paths.

Once a flow is *marked*, its rate is fixed.

The flowchart of *PFOR* is shown in Figure 3.1. The algorithm is described as follows.

Procedure *PFOR*

A. ***Initialization***.

All flow rates are set to 0 and no flows are *marked*.

B. ***Progressive filling with optimal routing***

We increase all *unmarked* flows with equal rates while finding the optimal routes and flow partitions for them such that the rates and the total throughput are maximized simultaneously. Then the algorithm goes to *Step C*.

Note: *Step B* stops when either one of the following conditions holds:

- A flow gets the bandwidth equal to its demand.
- A flow meets bottleneck links through all its feasible paths.

C. ***Are there bottleneck links?***

In this step, we check the current state of the network. If a flow gets the bandwidth equal to its demand, the algorithm goes to *Step D*. If a flow meets bottleneck links through all its feasible paths, the algorithm goes to *Step E*.

D. ***Mark min-demand flows***

We *mark* the *unmarked* flow(s) with minimum bandwidth demand. Then the algorithm goes to *Step F*.

Note: Although the rate of a *marked* flow is fixed, the partition of this flow is not fixed yet in order to provide maximum flexibility and the best allocation for other *unmarked* flows.

E. ***Mark bottleneck-crossing unmarked flows***

We mark all *unmarked* flows which meet bottleneck links through all their feasible paths. Then the algorithm goes to *Step F*.

F. ***Are all flows marked***?

We check whether all flows are *marked*. If all flows are *marked*, the algorithm goes to *Step G*. Otherwise, the algorithm returns to *Step B* to repeat the process.

G. ***Min Aggregate Link Loading***

To improve the bandwidth utilization efficiency further, we reroute all *marked* flows with the objective of minimizing the aggregate link loading. This objective is achieved by the following task.

Task. Minimize Aggregate Link Loading

Constants:

N	The number of nodes
$\mathbf{E} = \left[e_{ij} \right]_{N \times N}$	The *marked* rate matrix.

C_i The capacity of ith link in the outer ring, $i = 0,1,\cdots,N-1$

D_i The capacity of ith link in the inner ring, $i = 0,1,\cdots,N-1$

Variables:

$\boldsymbol{\alpha} = \left[\alpha_{ij}\right]_{N\times N}$ Percent of a rate used in the outer ring, $i, j = 0,1,\cdots,N-1$

u_i The aggregate load of the ith link in the outer ring, $i = 0,1,\cdots,N-1$

v_i The aggregate load of the ith link in the inner ring, $i = 0,1,\cdots,N-1$

Minimize:

$$L(\boldsymbol{\alpha}, \mathbf{u}, \mathbf{v}) = \sum_{i=0}^{N-1} u_i + \sum_{i=0}^{N-1} v_i \tag{3.1}$$

Subject to constraints:

$$\begin{aligned}
&\sum_{j=(i+1)_{\bmod N}}^{(i-1)_{\bmod N}} \alpha_{ij} e_{ij} + \sum_{k=(i+2)_{\bmod N}}^{(i-1)_{\bmod N}} \sum_{j=(i+1)_{\bmod N}}^{(k-1)_{\bmod N}} \alpha_{kj} e_{kj} - u_i = 0 \\
&\sum_{j=(i+1)_{\bmod N}}^{(i-1)_{\bmod N}} (1-\alpha_{ij}) e_{ij} + \sum_{k=(i+1)_{\bmod N}}^{(i-2)_{\bmod N}} \sum_{j=(k+1)_{\bmod N}}^{(i-1)_{\bmod N}} (1-\alpha_{kj}) e_{kj} - v_i = 0
\end{aligned} \tag{3.2}$$

and

$$\begin{aligned}
&0 \le \alpha_{ij} \le 1, && \text{for } i, j = 0,1,\cdots,N-1 \text{ and } i \ne j \\
&\alpha_{ij} = 0, && \text{for } i, j = 0,1,\cdots,N-1 \text{ and } i = j
\end{aligned} \tag{3.3}$$

$$\begin{aligned}
&0 \le u_i \le C_i, && \text{for } i = 0,1,\cdots,N-1 \\
&0 \le v_i \le D_i, && \text{for } i = 0,1,\cdots,N-1
\end{aligned} \tag{3.4}$$

END

When *PFOR* terminates, max-min fair allocation and optimal routing and flow partitioning problems are solved jointly. The proof can be found in [20].

Next, we illustrate *PFOR* by a simple example. We consider the network of Figure 3.2. This packet ring network consists of 4 nodes and 8 links. The outer ring routes packets clockwise and the inner ring routes packets anticlockwise. We assume that all link capacities are the same, which equal 100 units per second. There are four active flows with bandwidth demands, $\gamma_{02} = 120$, $\gamma_{03} = 20$, $\gamma_{12} = 90$ and $\gamma_{32} = 40$ units per second, respectively.

Initially, all flows are *unmarked* and with rates set to 0. The algorithm enters *Step B*. By optimal routing, this step tries to maximize the total throughput under the constraint that each *unmarked* flow rate must increase at the same pace. Upon entering *Step C*, it is found that there is no bottleneck link. At *Step D*, since only flow γ_{03} gets a rate equal to its demand, flow γ_{03} is *marked.* As shown in Figure 3.3, the other flows get the same rate as flow γ_{03}.

Since not all flows are *marked*, the algorithm returns to *Step B* and repeats the process. At *Step B*, the rate of flow γ_{03} will not increase any more and its rate is regarded as a constant, because it has already been *marked.* All other flows increase their rates at the same pace. After *Step B*, *Step C* finds that there still is no bottleneck link and the algorithm goes to *Step D*. At *Step D*, flow γ_{32} is *marked*, because it has the minimum bandwidth demand among all *unmarked* flows. As shown in Figure 3.4, the other *unmarked* flows get the same rate as flow γ_{32}.

Flows γ_{12} and γ_{02} are still not *marked*, so the algorithm returns to *Step B* to repeat the process. *Step B* increases their rates at the same pace. After *Step B* is processed, *Step C* finds that both *unmarked* flows meet bottleneck links through all their feasible paths. The bottleneck links for flow γ_{12} is link (1, 2) on the outer ring and link (3, 2) on the inner ring. 20 units of traffic of flow γ_{12} go through the outer ring while 60 units go through the inner ring. All traffic of flow γ_{02} goes through only the outer ring. The bottleneck links for flow γ_{02} are link (1, 2) on the outer ring. Then both flows are *marked* and they get an equal rate of 80 units per second. The results are shown in Figure 3.5. *Step F* finds that all flows are marked, so the algorithm goes to *Step G*.

Now, flows γ_{03}, γ_{32}, γ_{12} and γ_{02} get rates of 20, 40, 80 and 80 units per second, respectively. The total throughput is 220 units per second. We note that the solutions of max-min fair allocation and the total throughput are unique while the solution of flow partitioning may not be unique. In other words, we may have different choice to maintain max-min fairness in some special cases. Another flow partitioning to maintain the same max-min fair allocation for this example is shown in Figure 3.6. Flow γ_{12} only goes through the outer ring and gets a rate of 80 units per second. Flow γ_{02} goes through two rings. 20 units of traffic go through the outer ring, while 60 units go through the inner ring. This example tells us that we still have some flexibility to control the traffic flow partitioning while maintaining max-min fair allocation with optimal routing. Note that these two flow partitioning solutions give different results of the aggregate link loading. For example, the aggregate link loading for the case shown in Figure 3.5 is 420 units per second, while 300 units per second for the case shown in Figure 3.6. In order to improve the bandwidth utilization efficiency further, the algorithm reroutes all the marked flows with the objective of minimizing the aggregate link loading at *Step G*. The final max-min fair allocation with optimal routing solution is shown in Figure 3.6.

To compare, we also illustrate *PF* with the shortest path routing by this example. Since flow γ_{02} has two shortest paths, we assume flow γ_{02} chooses the outer ring paths without loss of generality. For flow γ_{03} and γ_{32}, the results are the same by using *PF*. However, only 50 units per second are obtained by flow γ_{02} and γ_{12}. The total throughput is only 160 units per second. Compared with *PFOR*, about 27 percent throughput is sacrificed. This is shown in Figure 3.7.

This example shows the amount of throughput is increased by optimal routing. Max-min fair allocation with the shortest path routing only allows one route for a flow and when this route is congested, there is no way to route the traffic to an alternative path which is congestion-free. By using optimal routing, max-min fair allocation improves the throughput performance by using one or more feasible paths simultaneously.

At last, we briefly analyze the complexity of *PFOR* algorithm. As we mentioned before, *PFOR* algorithm terminates in a finite time, because the number of links and flows are finite. Since there are totally *N*(*N*-1) flows, PFOR algorithm runs at most *N*(*N*-1) loops, where each loop consists of 4 steps (Step B, C, D, F or Step B, C, E, F). This is the worst case that happens when the network is lightly loaded and there is no packet loss. Within each loop, we have to solve a Maximum Flow problem (Step B). As analyzed in the previous chapter, the maximum throughput routing problem has the complexity $O\left(N^3\right)$ for a packet ring network. So the worst case complexity for *PFOR* algorithm is $O\left(N^5\right)$. Therefore, *PFOR* algorithm can be solved in polynomial time. If the network is under extremely high offered load (suppose each flow has infinite bandwidth demand), *PFOR* algorithm terminates within one loop. In this extreme case, all flows will be *marked* within one loop. In this case, the algorithm terminates no more than $O(N^3)$ time. Therefore, depending on the traffic patterns, the run times of *PFOR* algorithm may be different.

In a later section, we compare the throughput performance between max-min fair allocation with optimal routing and max-min fair allocation with shortest path routing by more general numerical results.

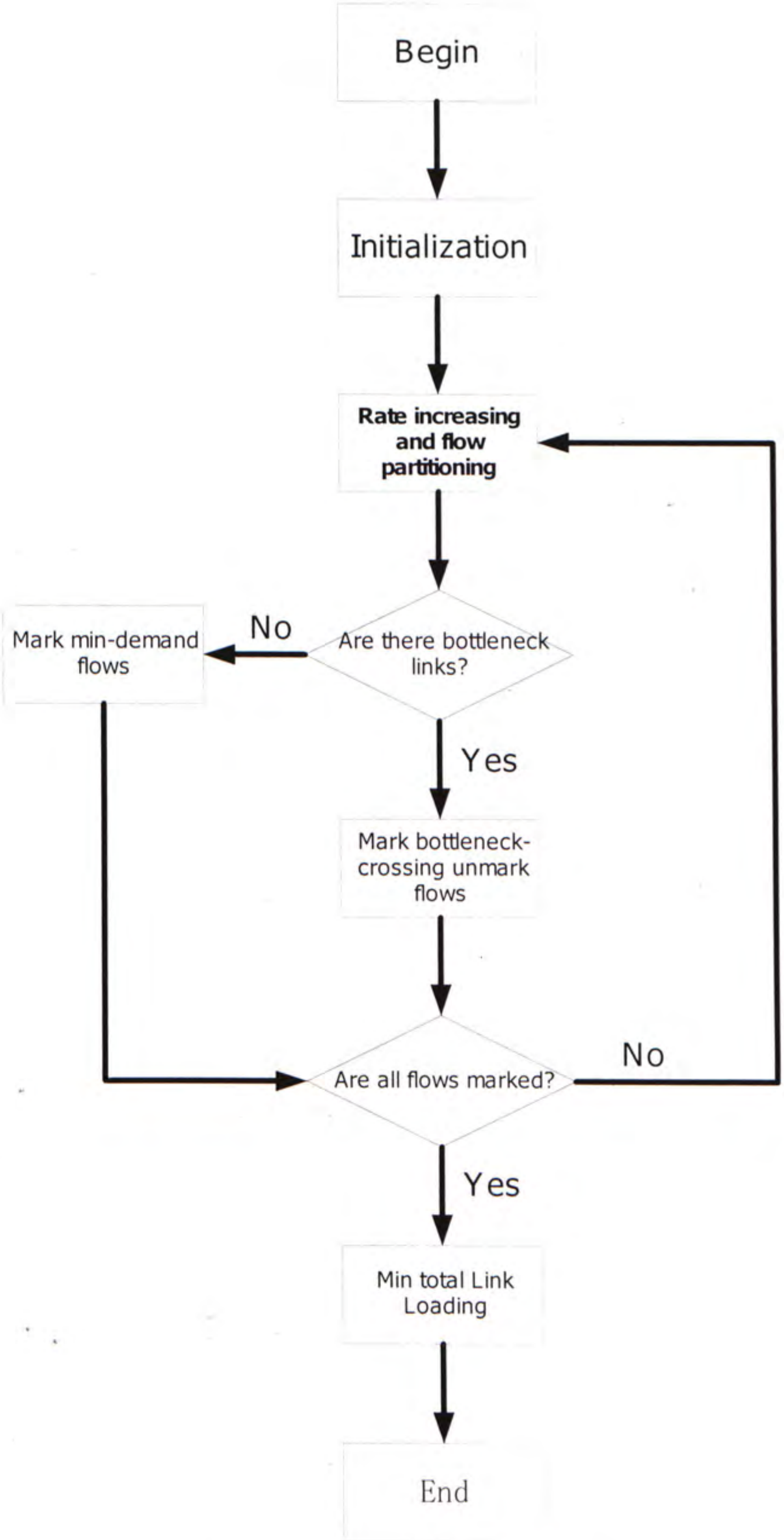

Figure 3.1 The flowchart of *PFOR*

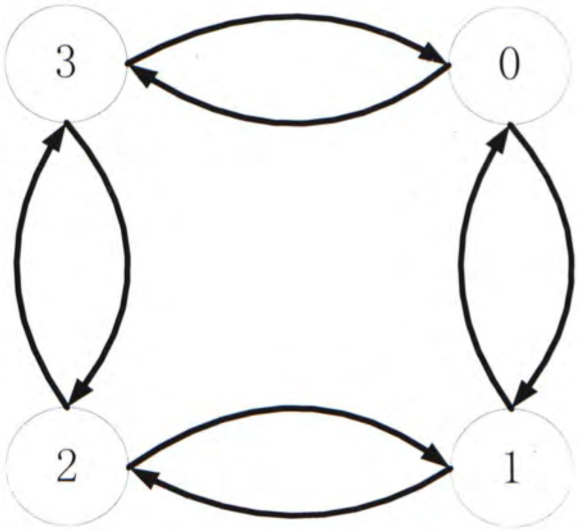

Figure 3.2 4-node packet ring network example

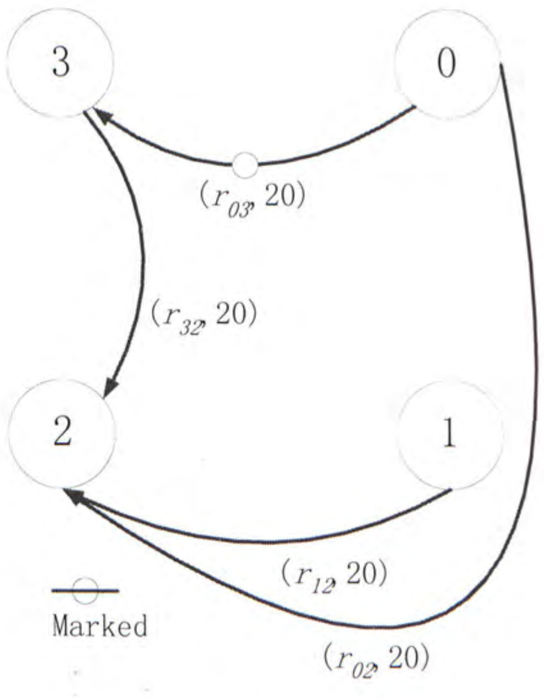

Figure 3.3 Step 1 of *PFOR* example

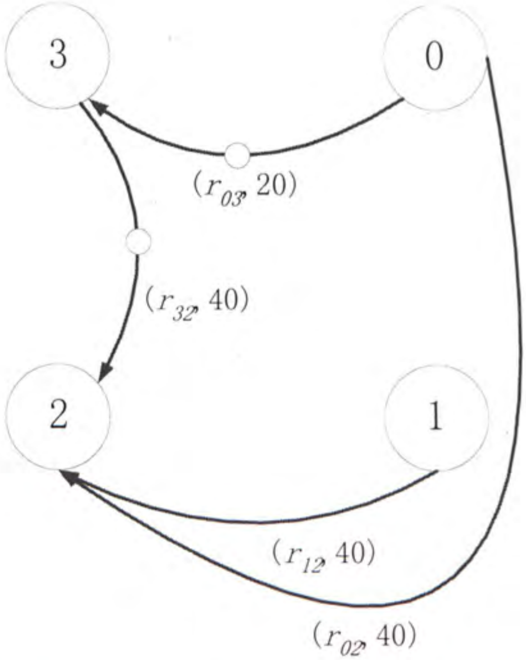

Figure 3.4 Step 2 of *PFOR* example

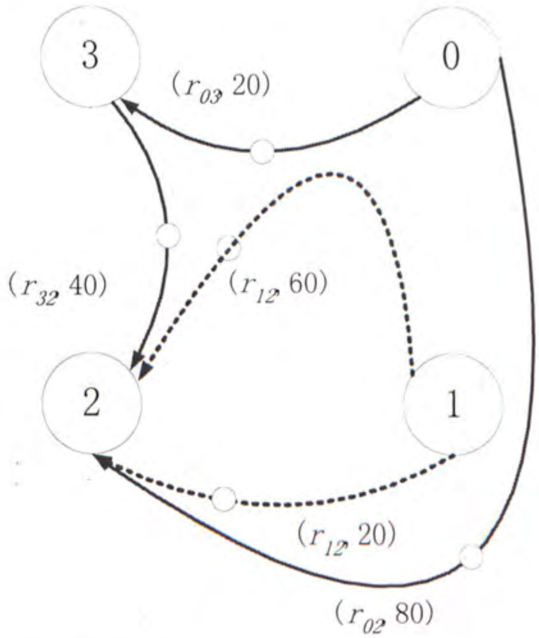

Figure 3.5 Step 3 of *PFOR* example

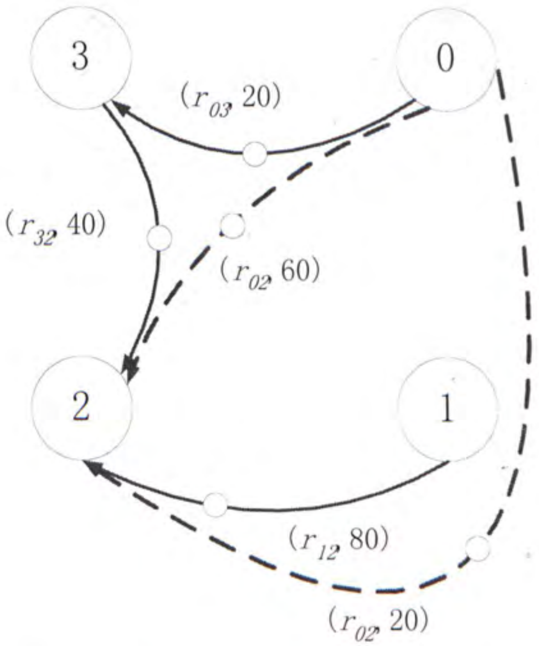

Figure 3.6 Another flow partitioning solution of *PFOR* example

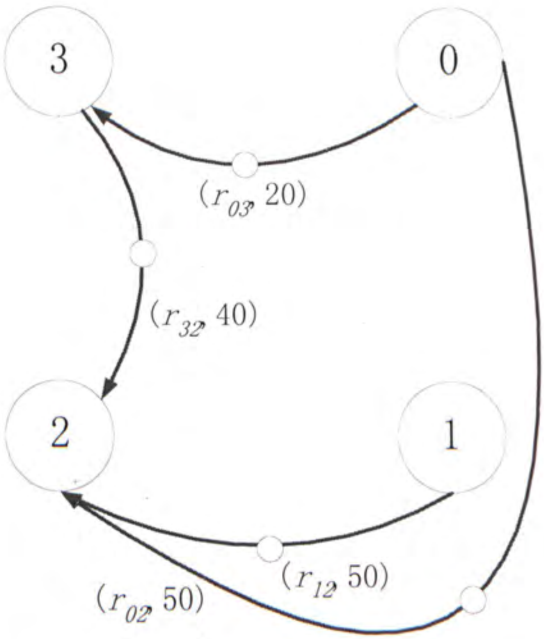

Figure 3.7 An example of *PF* with shortest path routing

3.3 Proportionally Fair Allocation

As described in Theorem 1.3.2 in Chapter 1, proportionally fair allocation is coupled with utility optimization problem. The utility is defined as a logarithmic function of bandwidth under the elastic traffic assumption [15]. Proportionally fair allocation is the unique solution of this aggregate utility optimization problem. The definition of proportional fairness directly extends to the case where each flow has multiple routes. We argue the viewpoint as follows. By using optimal routing instead of shortest path routing, both the aggregate utility performance and the throughput performance can be improved. The proportionally fair allocation for a packet ring network is formulated as follows.

Task. Proportionally Fair allocation

Constants:

Same as before.

Variables:

Same as before.

Maximize:

$$L(\mathbf{x},\mathbf{z}) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\log\left(x_{ij}\right) \tag{3.5}$$

Subject to constraints:

Same as before.

The objective function (3.5) is differentiable and strictly concave. The feasible region is compact. So the local maximum is the global maximum [15]. We assume that no flow is allowed to get more bandwidth than its demand. Reference

[15] gives the analytical solution of this problem by Lagrangian methods [34] under the assumption that a single route is used by a flow. For proportionally fair allocation with optimal routing case, it is difficult to express the analytical solution in an explicit form. In addition, it is difficult to find a heuristic algorithm, such as *Progressive Filling* (*PF*) algorithm used in max-min fair allocation, to find the solution of proportionally fair allocation. Proportionally fair allocation can only be obtained by solving the aggregate utility optimization problem. A general convex programming problem can be solved by Barrier method [47]. Reference [47] also gives the complexity analysis of this method.

In Section 3.4, we investigate the aggregate utility performance and the throughput performance for proportionally fair allocations with optimal routing and that with shortest path routing, respectively.

3.4 Numerical Results

We consider the network shown in Figure 3.8. This network consists of 4 nodes and 8 links. We also assume that all link capacities are the same, which equal 200 units per second. The traffic patterns are the same as before. The results shown here are the average values based on ten experiments with different traffic matrices.

First, we investigate the total throughput as a function of offered load for max-min fair allocations with shortest path routing and that with optimal routing. For comparison, we also draw the throughput performance of the maximum throughput routing. The numerical results are shown in Figure 3.9. When the network is lightly loaded, all three algorithms give similar throughput. When the network is under high offered load, max-min fair allocation with optimal routing gives a better throughput performance than that with shortest path routing. However, both max-min fair allocations with optimal routing and with shortest path routing show poorer throughput performance than the maximum throughput routing. Next, we illustrate the results more specifically. When the offered load index is equal to 1, all algorithms show the same throughput about 594 units per second, since the network is lightly loaded and there is no packet loss. For the point that the offered load index is 3, throughput for max-min fair allocation with shortest path routing, max-min fair allocation with optimal routing and the maximum throughput routing are 998, 1065 and 1240 units per second, respectively. Max-min fair allocation with optimal routing increases throughput by about 7%, compared with max-min fair allocation with shortest path routing. Compared with the maximum throughput routing, max-min fair allocation with optimal routing has about 14% lower throughput and max-min fair allocation with shortest path routing has about 20% lower throughput. When the offered load is extremely high, the throughput improvement of max-min fair allocation with optimal routing becomes insignificant.

Second, we investigate the throughput performance as a function of offered load for proportionally fair allocation with shortest path routing and that with optimal routing. For comparison, we also show the total throughput of the maximum throughput routing. The numerical results are shown in Figure 3.10. When the network is lightly loaded, all three algorithms give similar throughput. Under high offered load, proportionally fair allocation with optimal routing offers significant throughput gain over proportionally fair allocation with shortest path routing. Specifically, when the offered load index is equal to 1, all allocations show the same throughput about 594 units, since the network is lightly loaded and there is no packet loss. When the offered load index is increased to 8, the throughput for proportionally fair allocation with the shortest path routing, optimal routing and the maximum throughput routing are 1210, 1309 and 1462 units per second, respectively. In other words, proportionally fair allocation with optimal routing increases throughput by about 8%, compared to that with shortest path routing.

Next, we compare the aggregate utility performance for proportionally fair allocation with shortest path routing and that with optimal routing. Figure 3.11 shows that when the offered load index is 1, both of them achieve an aggregate utility value of 45, since the network is lightly loaded and there is no packet loss. When the offered load index is increased to 8, the aggregate utilities of proportionally fair allocation with optimal routing and that with shortest path routing are about 56 and 54, respectively. Therefore, proportionally fair allocation with optimal routing improves the aggregate utility than that with the shortest path routing.

We are also interested in finding the scalability property by optimal routing for both max-min fair allocation and proportionally fair allocation as a function of the number of nodes. The network parameter configurations and the traffic patterns are the same as before (in Chapter 2). We assume that each network is under high offered load. The result in Figure 3.12 shows that the throughput gain for max-min fair allocation by optimal routing becomes less when the network size becomes larger. On the contrary, the larger the network size, the better is the throughput improvement by optimal routing for proportionally fair allocation.

To summarize, in order to maintain max-min fairness, throughput must be sacrificed. Max-min fair allocation with optimal routing gives less throughput degradation than that with shortest path routing. Similarly, to maintain proportional fairness, throughput must also be sacrificed. Proportionally fair allocation with optimal routing also requires less throughput sacrifice than that with shortest path routing. In addition, proportionally fair allocation with optimal routing gives a better aggregate utility performance than that with shortest path routing. The throughput improvement for max-min fair allocation by optimal routing becomes less when a packet ring network size becomes larger. While the throughput improvement for proportionally fair allocation by optimal routing becomes larger when a packet ring network size becomes larger.

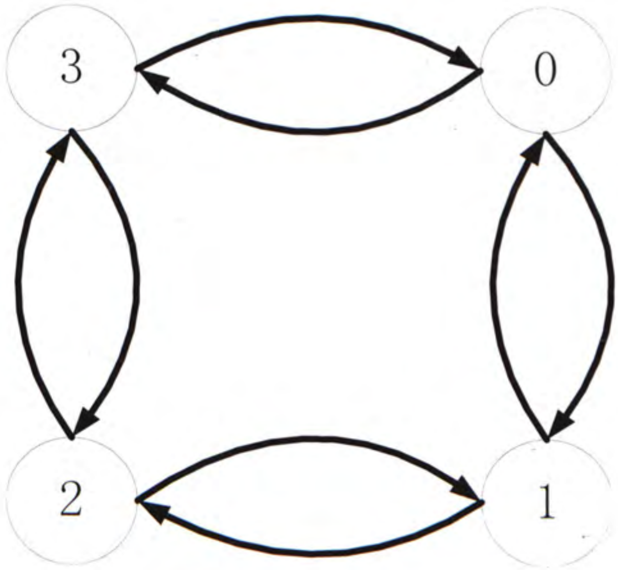

Figure 3.8 4-node packet ring network

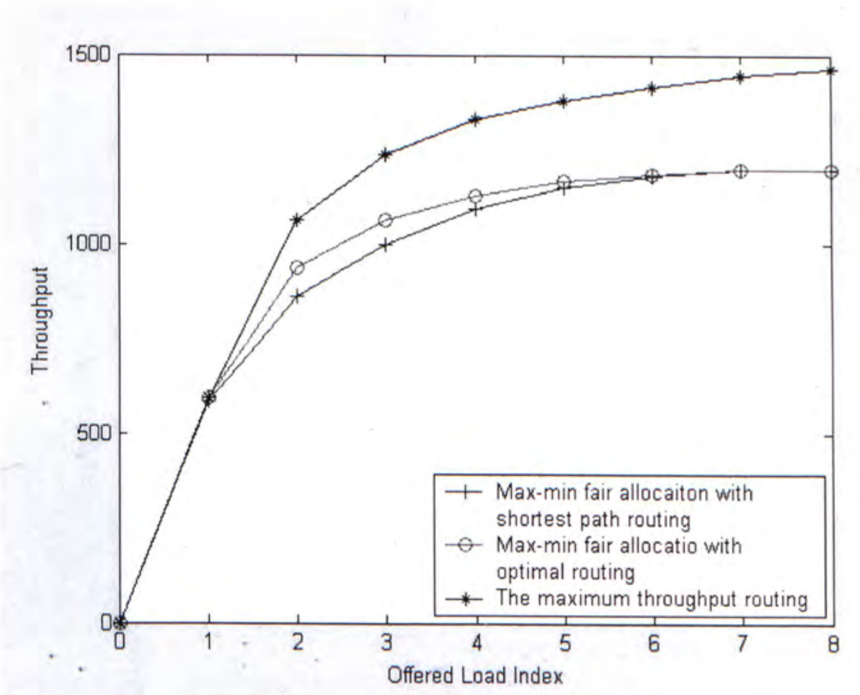

Figure 3.9 Throughput vs. offered load for max-min fair allocation

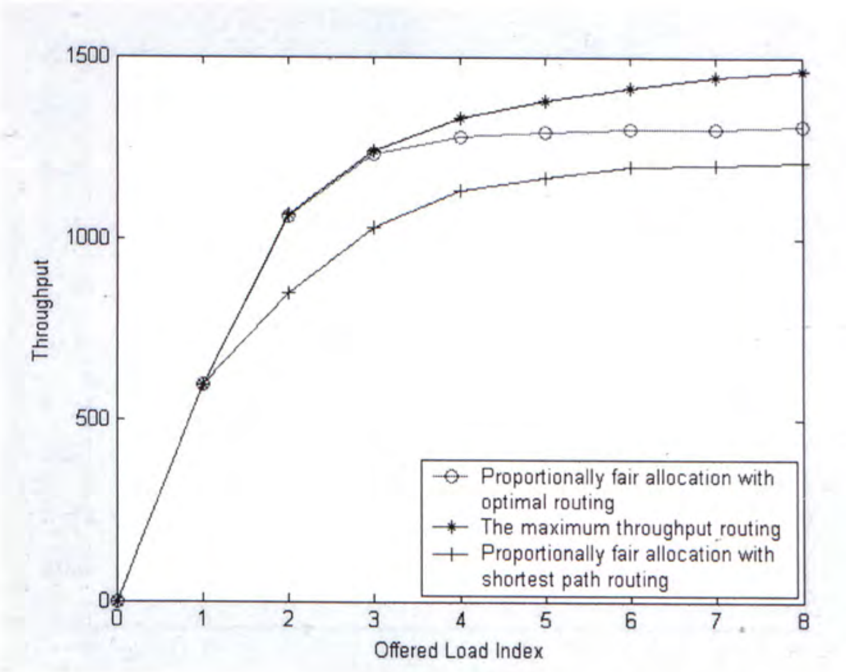

Figure 3.10 Throughput vs. offered load for proportionally fair allocation

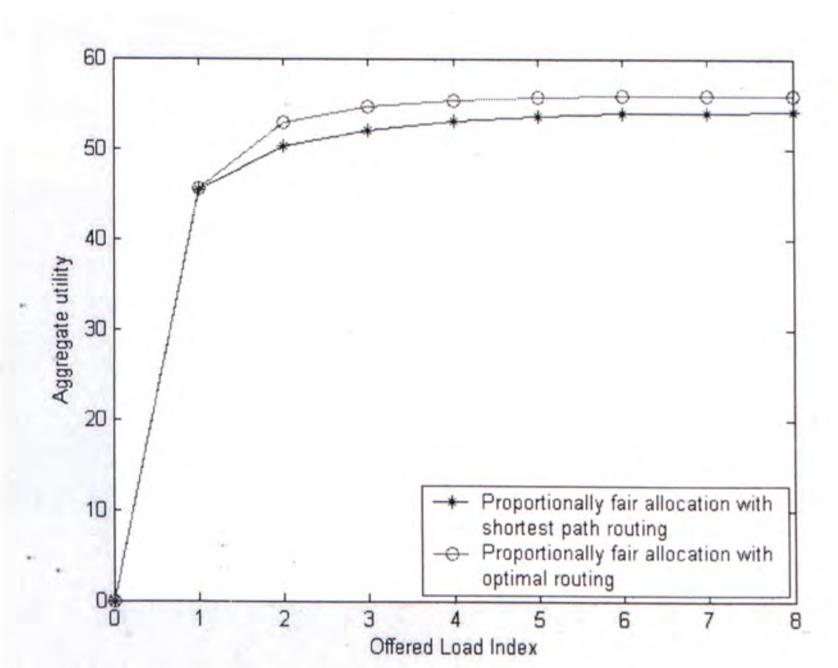

Figure 3.11 Utility vs. offered load for proportionally fair allocation

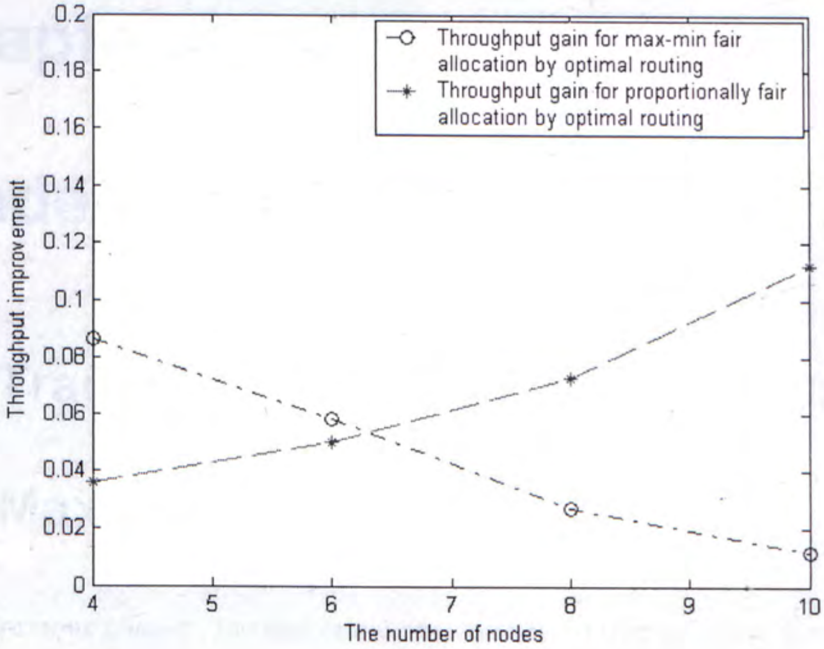

Figure 3.12 Throughput improvement for max-min fair allocation and proportionally fair allocation over optimal routing

Chapter 4

Tradeoff Analysis

4.1 Tradeoff between Throughput and Max-min Fairness

In the previous chapter, the numerical results showed that max-min fair allocation was achieved at the expense of sacrificing the total throughput. From the point of view of a network operator, it is undesirable to sacrifice the total throughput too much to maintain the perfect max-min fair allocation. What we are interested here is to find a flexible way that can provide tradeoff between the total throughput and the max-min fairness. The following task is designed to achieve this objective.

Task. Throughput-Fairness Tradeoff

Constants:

b A tuning parameter

$\mathbf{H} = \left[h_{ij} \right]_{N \times N}$ The rate matrix of max-min fair allocation with optimal routing

Others are the same as before.

Variables:

$\mathbf{K} = \left[k_{ij} \right]_{N \times N} = \left[\dfrac{x_{ij}}{h_{ij}} \right]_{N \times N}$ The normalized rate matrix (normalized to max-min fair allocation rate)

Others are the same as before.

Maximize:

$$L(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{ij} \tag{4.1}$$

Subject to constraints:

$$\left(k_{ij} - 1\right)^2 \le b \qquad \text{for all } i, j \tag{4.2}$$

Others are as same as before.

In this task, we control the amount of tradeoff between throughput and max-min fairness by adjusting the tuning parameter b. Constraint (4.2) shows that the tuning parameter b is used to limit the deviation of a normalized flow rate from one. Since a normalized flow rate is bounded within the range $\left[0, \min\left(\dfrac{\gamma_{ij}}{h_{ij}}, \dfrac{C}{h_{ij}}\right)\right]$, the value of the tuning parameter b is bounded within the range $\left[0, \max\left[\min\left(\dfrac{\gamma_{ij}}{h_{ij}}, \dfrac{C}{h_{ij}}\right) - 1\right]^2\right]$, where γ_{ij} is the demand of a flow and C is the link capacity. When the parameter b is 0, this task degenerates to max-min fair allocation with optimal routing. When the parameter b is at its upper bound, this task becomes the maximum throughput routing problem. One point is that the upper bound of b depends on the demand. Practically, we do not want a flow rate to deviate too much from its max-min fair rate. So we set b to a small value to control this deviation. In the next section, we will investigate how the tuning parameter b impacts the max-min fairness and the total throughput.

4.2 Numerical Results

In this section, we continue to use the network model shown in Figure 3.8. The link configuration and the traffic pattern are both as same as before. The results shown here are the average values based on ten experiments with different traffic matrices.

In Figure 4.1, we show the total throughput as a function of the offered load index given different values of b. The result of max-min fair allocation (b=0) gives the lower bound for throughput, while the result of the maximum throughput routing gives the upper bound for throughput (b=0.75) in this case. Figure 4.1 shows that the throughput is improved with the increase of b.

In Figure 4.2, we show the corresponding max-min fairness value. Here, maximum throughput routing (b=0.75) gives the lower bound of the fairness index, while max-min fair allocation with optimal routing (b=0) gives the upper bound of the fairness index. Figure 4.2 shows that the fairness index drops with the increase of b.

To see the impact of the total throughput and the fairness index against the tuning parameter b more clearly, we compare the throughput and the fairness index as a function of the normalized parameter b in Figure 4.3 and Figure 4.4, respectively. In Figure 4.3, when the parameter b is increased within a small range near the origin, the total throughput increases a lot. Beyond that the throughput is not very sensitive to b. In Figure 4.4, the fairness index decreases with increasing b. The lowest fairness index is achieved when b is 0.75 in this case.

Figure 4.5 compares the throughput gain as the function of the fairness sacrifice. The throughput performance can be improved while sacrificing the fairness performance little only when the tuning parameter b is changed within a small range near zero. When b becomes large, the throughput performance cannot be improved much by sacrificing the fairness performance. In other words, a good tradeoff between the total throughput and the max-min fairness can only be achieved by setting the tuning parameter b small. With similar scenario, we also investigate the tradeoff between throughput and max-min fairness for another 8-node packet ring network. The tradeoff is better than that for 4-node case.

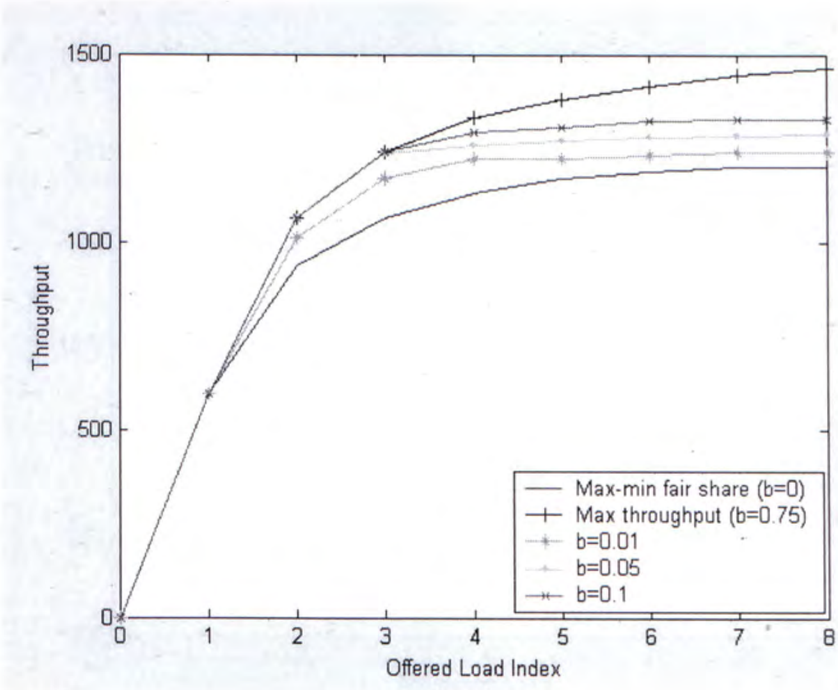

Figure 4.1 Throughput vs. offered load

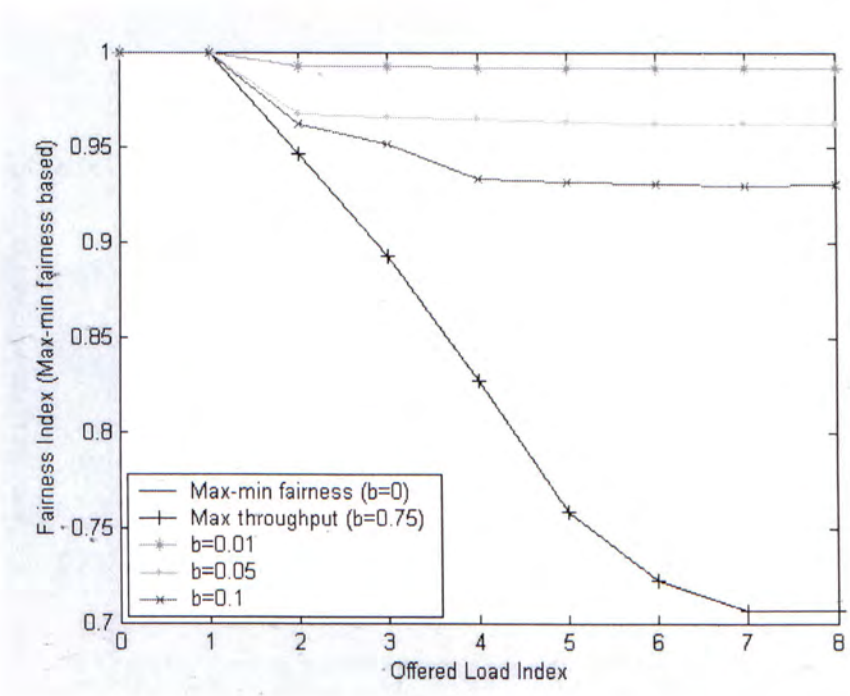

Figure 4.2 Fairness Index vs. offered load

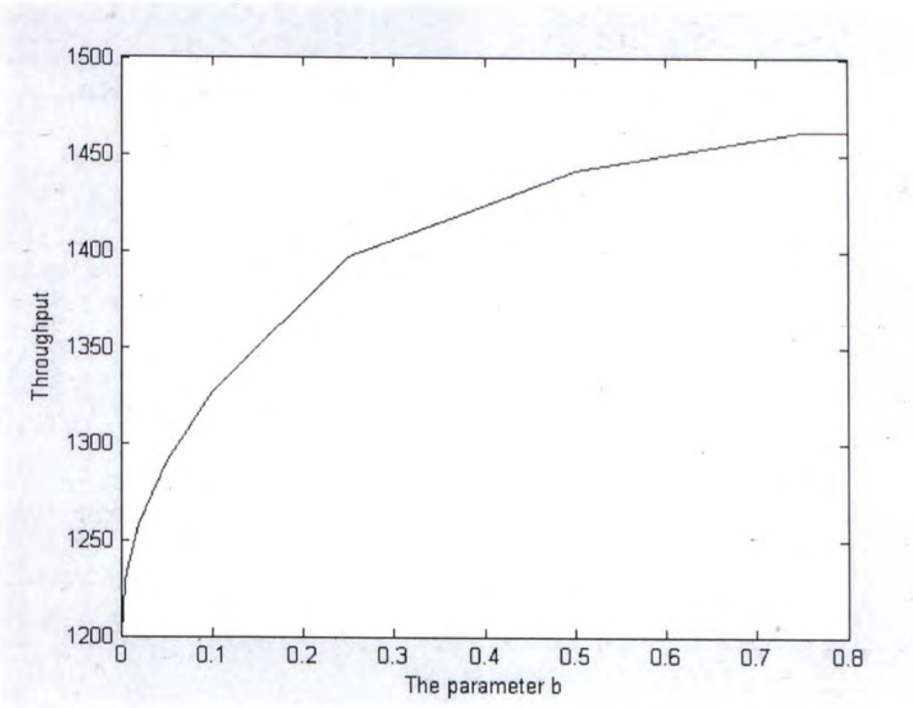

Figure 4.3 Throughput vs. the parameter b

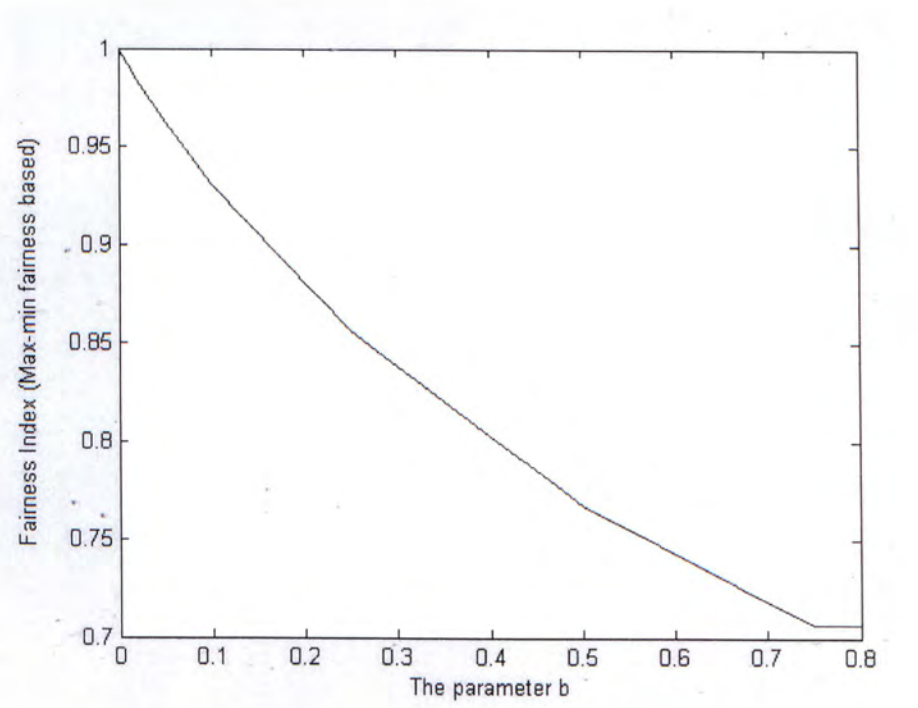

Figure 4.4 Fairness index vs. the parameter b

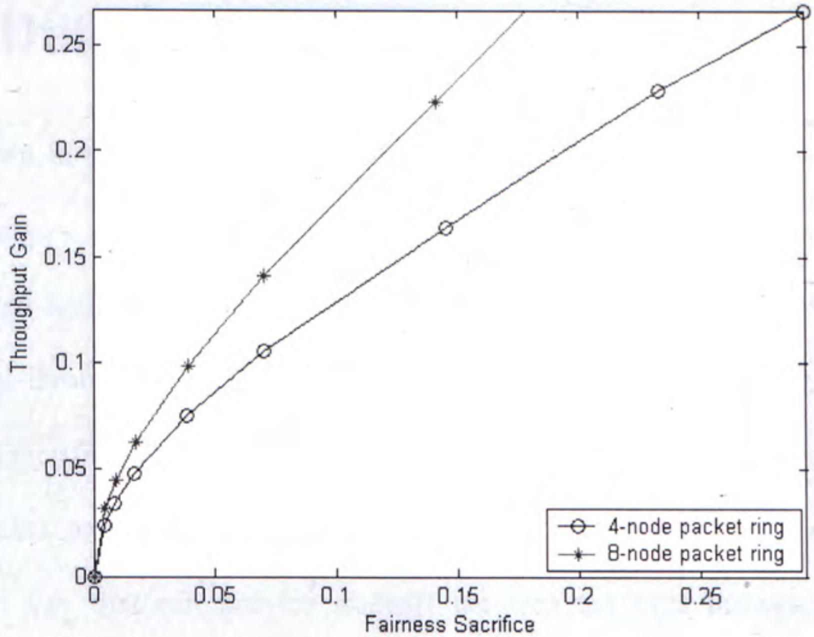

Figure 4.5 Throughput gain vs. fairness sacrifice

4.3 Tradeoff between Throughput and Utility

As shown in the previous chapter, proportionally fair allocation is coupled with the aggregate utility optimization problem. The numerical results showed that the optimized aggregate utility performance is achieved at the expense of sacrificing the total throughput. From the point of view of a network operator, we do not hope to sacrifice throughput much to maintain the optimized aggregate utility (i.e. the perfect proportionally fair allocation). We are interested in looking for a flexible way that can provide tradeoff between the total throughput and the aggregate utility. Thus, when the aggregate utility performance is not as important as the throughput performance for a system, we can arbitrarily make a choice that how much sacrifice of the aggregate utility performance we can afford in order to gain the improvement of the total throughput performance. This is the basic idea of the following task.

Task. Throughput-Utility Tradeoff

Constants:

b A normalized parameter, $0 \le b \le 1$

Others are the same as before.

Variables:

Same as before.

Maximize:

$$L(\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left[(1-b)\log(x_{ij}) + bx_{ij}\right] \tag{4.3}$$

Subject to constraints:

Same as before.

In this task, the objective function consists of the aggregate utility term and the total throughput term. The tradeoff between the aggregate utility and the total throughput is controlled by parameter b. The objective function is strictly concave and continuous differentiable function over its rate range. The feasible region is convex and compact. So the unique optimal solution exists. In next section, we will investigate how the normalized parameter affects the aggregate utility and the total throughput.

4.4 Numerical Results

In this section, we use an 8-node packet ring network model. The link capacity is 1000 units.

In Figure 4.6, we compare the total throughput as the function of the offered load given different values of b. The result of proportionally fair allocation (b=0) gives the lower bound for throughput, while the result of the maximum throughput routing (b=1) gives the upper bound for throughput. The throughput performance is improved with increasing the normalized parameter b from 0 to 1. To illustrate, we consider the point that the offered load index is equal to 9. When b is increased from 0 to 0.005, the total throughput is changed from 9648 to 10060 units and increased by about 4%. However, if b is increased continuously, the larger b, the lower increasing speed of throughput. The increase of throughput becomes insignificant when b is getting close to 1.

Correspondingly, we show the impact to the aggregate utility of different values of b in Figure 4.7. In this case, the result of the maximum throughput routing (b=1) gives the lower bound of the aggregate utility, while the result of proportionally fair allocation (b=0) gives the upper bound of the aggregate utility. The aggregate utility performance drops with increasing b. We consider the point that the offered load index is equal to 9. When b is increased from 0 to 0.005, the aggregate utility is changed from 267.06 to 265.75 and decreased by only 0.5%. With increasing b continuously, the larger b, the faster decreasing speed of the aggregate utility. The dropping of the aggregate utility will become insignificant when b is getting close to 1.

To see the impact of the total throughput and the aggregate utility against the normalized parameter b more clearly, we compare the throughput and the aggregate utility as the function of the normalized parameter b in Figure 4.8 and Figure 4.9, respectively. In Figure 4.8, when b is increased within a small range near the origin, say [0, 0.1], the total throughput increases very fast. Out of this range, the closer to 1 of b, the slower increasing speed of the total throughput. So only b is increased within a small range near the origin, can we gain throughput much while sacrificing the aggregate utility a little bit. This effect can be seen more clearly by combining Figure 4.8 and Figure 4.9 into Figure 4.10. With similar scenario, we also investigate the tradeoff between throughput and the aggregate utility for another 4-node packet ring network. The result shows that the tradeoff is not as good as that for the 8-node case.

At last, we conclude our results. The throughput performance can be improved while sacrificing the aggregate utility little only when the normalized parameter b is changed within a small range near zero. When b becomes close to 1, the throughput performance cannot be improved by sacrificing the aggregate utility. A good tradeoff between the total throughput and the aggregate utility can only be achieved by setting the normalized parameter b a small value. The tradeoff is better when the network becomes larger.

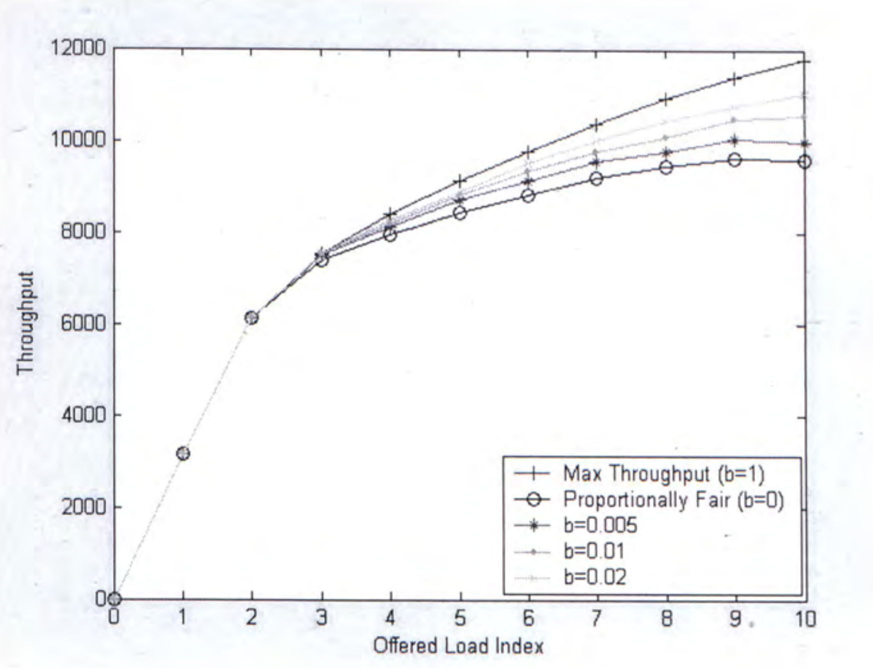

Figure 4.6 Throughput vs. offered load

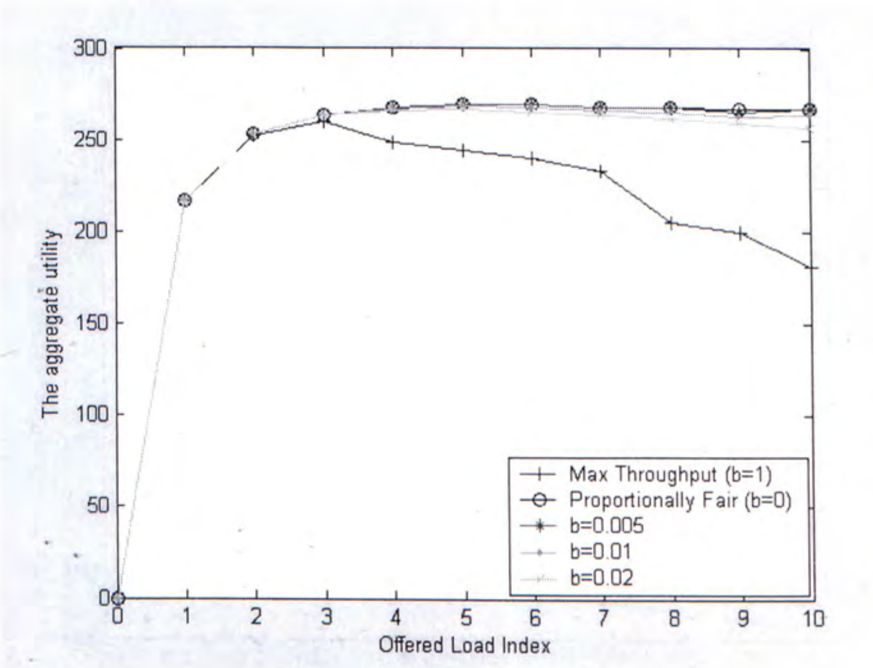

Figure 4.7 The aggregate utility vs. offered load

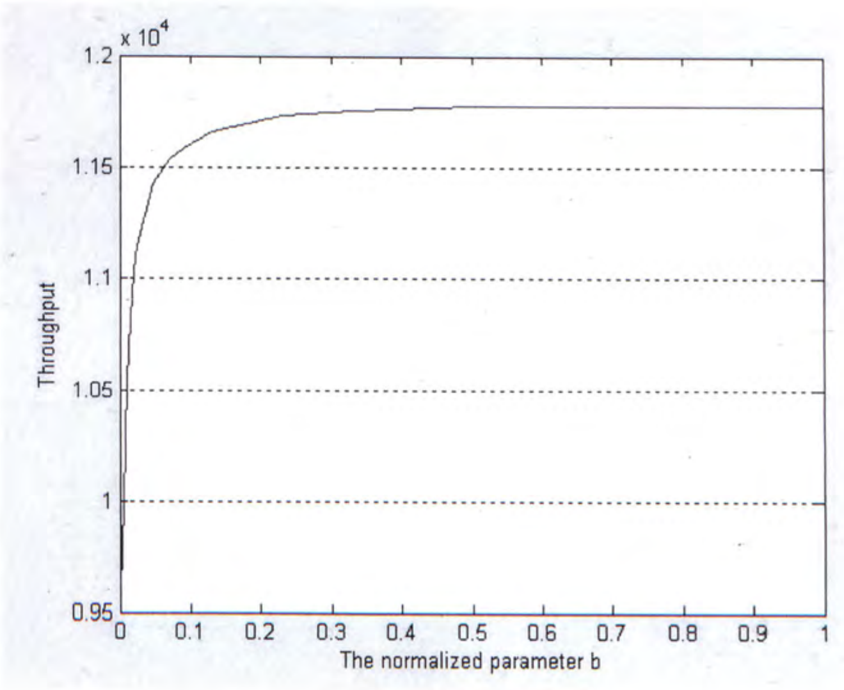

Figure 4.8 Throughput vs. b

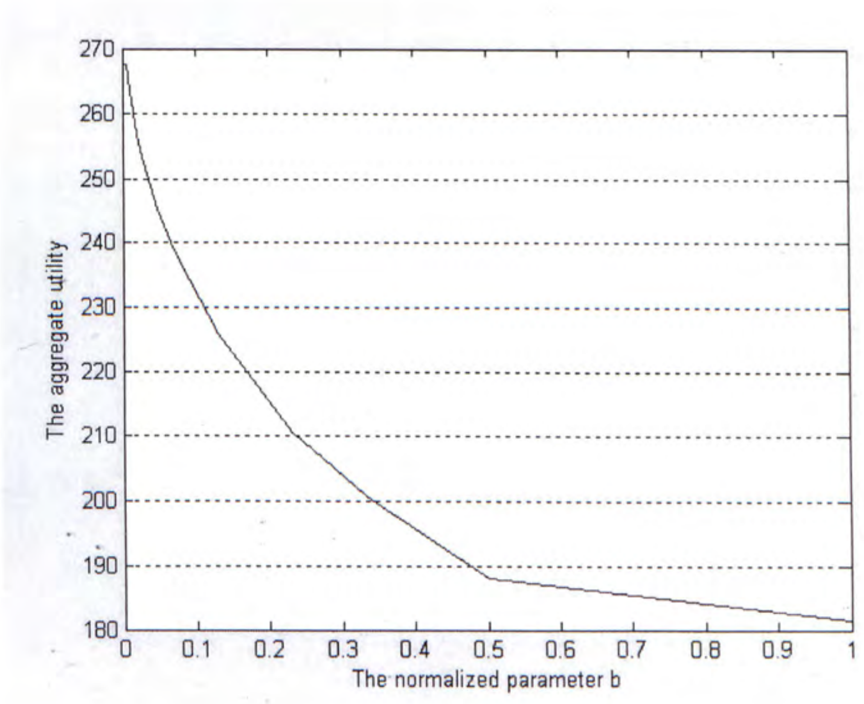

Figure 4.9 The aggregate utility vs. b

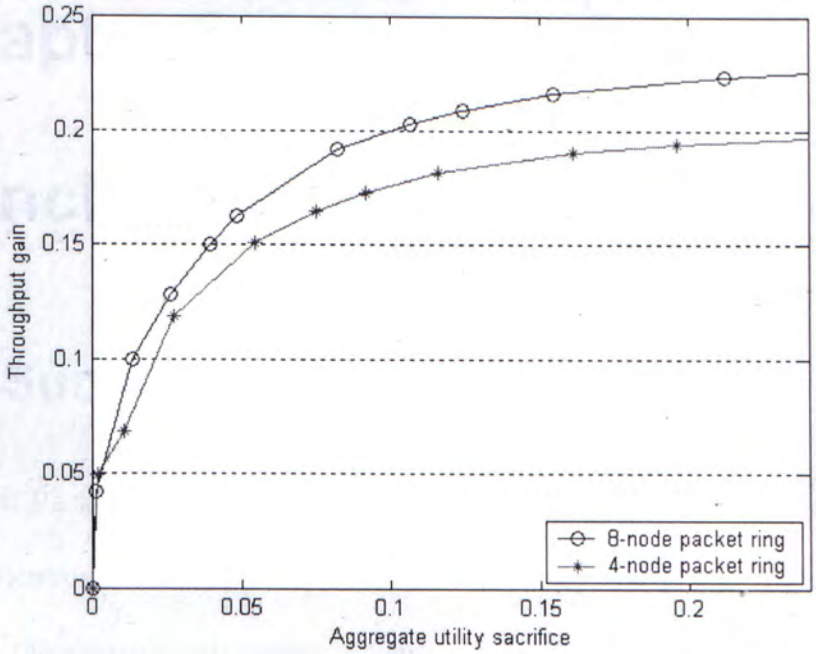

Figure 4.10 Throughput gain vs. the aggregate utility sacrifice

Chapter 5

Conclusion

5.1 Summary

Resilient Packet Ring (RPR), a new technology for metro network, is receiving more attention due to its good bandwidth efficiency and low cost properties. By default, the shortest path routing scheme is used for RPR. However, since traffic load may change dynamically and distribute unevenly, the simple shortest path routing is not necessarily efficient and fair to flows. This motivates us to look for new routing and flow control algorithms for RPR which can utilize bandwidth more efficiently while maintaining fairness.

In this thesis, we designed an optimal routing and flow control algorithm for maximizing throughput for a packet ring network. The numerical results showed that the maximum throughput routing had better throughput performance than the shortest path routing. The throughput gain became smaller when the network size became larger. The maximum throughput routing could not treat each individual flow fairly.

Then we studied fair bandwidth allocation problem for a packet ring network, based on two fairness criteria: max-min fairness and proportional fairness. To maintain max-min fairness under multiple routes assumption, we proposed a new algorithm called *Progressive Filling with Optimal Routing* (*PFOR*). *PFOR* solved max-min fair allocation and optimal routing jointly. Numerical results showed that *PFOR* had a better throughput performance than max-min fair allocation with shortest path routing. The throughput gain became less when the network size became larger. Similarly, proportionally fair allocation with optimal routing also gave a better throughput performance than that with shortest path routing. The throughput gain became larger when the network size became larger.

Finally, we studied two tradeoffs. One was between throughput and max-min fairness, the other was between throughput and the aggregate utility. For each case, we controlled the degree of the tradeoff by adjusting a tuning parameter. Numerical results showed that with proper choice of the parameter we could maintain near optimal throughput at only a little sacrifice of the max-min fairness or the aggregate utility. The tradeoffs were better when the network size became larger.

5.2 Discussion and Future Work

In this thesis, we proposed the *PFOR* algorithm to solve max-min fair allocation and optimal routing problems jointly. By *PFOR*, we can achieve max-min fair allocation while sacrificing throughput less than traditional max-min fair allocation under single route condition. Although we studied *PFOR* based on a packet ring network, *PFOR* is not designed only for packet ring networks. In the future, we will study the performance of the *PFOR* algorithm for a general mesh network.

During the course of studying, we found that to get the solution proportionally fair allocation is a time consuming job even for a small size packet ring network. To the best of our knowledge, there is no heuristic algorithm to solve proportionally fair allocation. By observing the characteristics of the proportionally fair solution, researchers found that the bandwidth allocation is approximately (inversely) proportional to the number of bottleneck links of a flow [15]. This observation gives us a hint that we may first derive a max-min fair solution and then approximate proportionally fair solution based on the above observation. To study heuristic algorithms to achieve proportionally fair allocation is another research direction.

Bibliography

[1] D. Tsiang and G. Suwala. "The Cisco SRP MAC Layer Protocol," *RFC* 2892, 2000.

[2] White Paper. *An Introduction to Resilient Packet Ring Technology*. Resilient Packet Ring Alliance. http://www.rpralliance.org/, 2001.

[3] White Paper. *Outline of the IEEE 802.17 RPR Draft Standard*. Resilient Racket Ring Alliance. http://www.rpralliance.org/, 2001.

[4] A.S. Tanenbaum. *Computer Networks*. 3rd ed. Prentice-Hall, 1996.

[5] W. Goralski. *Sonet*. 2nd ed. McGraw-Hill, New York, 2000.

[6] R. J. Wilson. *Introduction to graph theory*. 4th ed. Longman, Harloy, 1996.

[7] D. Bertsekas and R. Gallager. *Data Networks*. 2nd ed. Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

[8] S. Shenker. "Fundamental Design Issues for the Future Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1176-1186, Sep., 1995.

[9] R.F. Liao and A.T. Campbell. "Dynamic Edge Provisioning for Core IP Networks," *IWQOS. Eighth International Workshop on Quality of Service*, pp. 148-157, 2000.

[10] R. J. La and V. Anantharam. "Utility-Based Rate Control in the Internet for Elastic Traffic," *IEEE/ACM Transactions on Networking*, vol. 10, no. 2, pp. 272-286, April, 2002.

[11] L. Breslau and S. Shenker. "Best-effort versus Reservations: A Simple Comparative Analysis," *Proc. ACM SIGCOMM*, September, 1998.

[12] F.P. Kelly, A.K. Maulloo and D.K.H. Tan. "Rate Control for Communication Networks: Shadow prices, Proportional Fairness and Stability," *Journal of the Operational Research Society*, vol. 49, 1998.

[13] F.P. Kelly. "Mathematical modeling of the Internet," *Proceedings of the Fourth international Congress on Industrial and Applied Mathematics*, July, 1999.

[14] F.P. Kelly, S. Zachary and I. Ziedins. *Stochastic Networks: Theory and Applications*, volume 4. Oxford University Press, 1996.

[15] F.P. Kelly. "Charging and Rate Control for Elastic Traffic," *European Transactions on Telecommunications*, volume 8, 1997.

[16] H. Qingyanga, D.W. Peter. "Global Max-min Fairness Guarante for ABR Flow Control," *IEEE INFOCOM98*, 1998.

[17] J. Kleinberg, Y.Rabani and E. Tardos. "Fairness in Routing and Load Balancing," *IEEE Symposium on Foundations of Computer Science*, pp. 568-578, 1999.

[18] S. Vegesna. *IP Quality of Service*. Cisco Press. 2001.

[19] S. Keshav. *An Engineering Approach to Computer Networking: ATM Networks, the Internet and the Telephone Network*. Addison-Wesley, 1997.

[20] J.Y.L. Boudec. "Rate adaptation, Congestion Control and Fairness: A Tutorial," *Ecole Polytechnique Federale de Lausanne*, 2000.

[21] R. Jain, A. Durresi and G. Babic. "Throughput Fairness Index: An Explaination," *Technical Report 99-0045*, http://www.cis.ohio-state.edu/~jain/, 1999.

[22] R. Jain, W. Hawe and D. Chiu. "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," *DEC-TR-301*, September 26, 1984.

[23] D.M. Chiu and R. Jain. "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems*, 17, 1989.

[24] A.M. Collel, M.D. Whinston and J.R. Green. *Microeconomic Theory*. Oxford University Press, New York, 1995.

[25] C. Courcoubetis, F.P. Kelly and R. Weber. "Measurement-based Usage Charges in Communications Networks," *Operations Research*, 48, 2000.

[26] M. Schwartz. *Broadband Integrated Networks*. Prentice-Hall, Englewood Cliffs, New Jersey, 1996.

[27] A. Ortega and K. Ramchandran. "Rate-distortion Methods for Image and Video Compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23-50, Nov., 1998.

[28] L. Massoulie and J.W. Roberts. "Bandwidth Sharing: Objectives and Algorithms," *IEEE/ACM Transactions on Networking*, vol. 10, no. 3, pp. 320-328, June, 2002.

[29] Z. Youquan and F. Zhenming. "A New Fairness Criterion and Its Realization by Using a New Scheduling Algorithm in the Internet," *Sixth IEEE Symposium on Computers and Communications*, *Proceedings*, pp. 444-449, 2001.

[30] G. Malicsko, G. Fodor and M. Pioro. "Link Capacity Dimensioning and Path Optimization for Networks Supporting Elastic Services," *Communications*,

ICC 2002. *IEEE International Conference on* , Volume: 4, pp. 2304-2311, 2002.

[31] L. Kleinrock. *Queueing Systems*, volume 2. New York: Wiley, 1975.

[32] L. Kleinrock. *Queueing Systems*, volume 1. New York: Wiley, 1975.

[33] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, Massachusetts, 1997.

[34] Hillier and Lieberman. *Introduction to Operations Research*. 7th ed. McGraw-Hill, 2001.

[35] C. S. R. Murthy and M. Gurusamy. *WDM Optical Networks: Concepts, Design and Algorithms. Prentice-Hall*, New Jersey, 2002.

[36] R. Ramaswami and K.N. Sivarajan. *Optical Networks: A Practical Perspective*. Morgan Kaufmann, San Francisco, CA, 2002.

[37] R. Mazumdar, L.G. Mason and C. Douligeris. "Fairness in Network Optimal Flow Control: Optimality of Product Forms," *IEEE Transactions on Communications*, vol. 39, no. 5, pp. 775-782, May, 1991.

[38] T. Peter. *Next Generation Optical Networks: The Convergence of IP Intelligence and Optical Technology*. Prentice-Hall, New Jersey, 2002.

[39] American National Standards, "Token-passing Ring Access Methods and Physical Layer Specifications," *ANSI/IEEE 802.5*, 1985.

[40] American National Standard, "FDDI Token Ring Media Access Control (MAC)," *ANSI X3.139*, 1987.

[41] N. C. Strole, "Inside Token Ring version 2," *Data Communications*, pp. 117-125, January, 1989.

[42] I. Cidon and Y. Ofek, "MetaRing---A Full-Duplex Ring with Fairness and Spatial Reuse," *IEEE Transactions on Communications*, vol. 41, no. 1, pp. 110-120, January, 1993.

[43] M. C. Xu and J. H. Herzog, "Concurrent Token Ring Protocol," *Proceedings of IEEE INFOCOM'88*, U.S.A., pp. 145-154, 1998.

[44] G. Pacifici and A. Pattavina, "T-S Protocol: An Access Protocol for Ring Local Area Networks," *Proceedings of IEEE GLOBECOM'86*, U.S.A., pp. 25-28, 1986.

[45] P. C. Wong and T. S. Yum, "Design and Analysis of a Pipeline Ring Protocol," *IEEE Transactions on Communications*, vol. 42, no. 2, pp. 1153-1161, April, 1994.

[46] M. Y. Wee and A. E. Kamal, "A Patial-Destination-Release Strategy for the Multi-Token Ring Protocol," *Proceedings of Local Area Networks 17th*, pp. 639-648, 1992.

[47] S. Boyd and L. Vanlenberghe, *Convex Optimization*, Online book draft, http://www.stanford.edu/~boyd/cvxbook.html, December, 2002.

[48] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, 2nd ed., MIT Press, 2001.